Complexity of Finding Subgraphs with Prescribed Degrees and Pairwise-Distances

Complexity of Finding Subgraphs with Prescribed Degrees and Pairwise-Distances

Hiroshi ETO

Department of Systems Design and Informatics
Kyushu Institute of Technology

Abstract

This thesis deals with the problem of finding a maximum vertex-subset S of a given a graph such that the subgraph induced by S is r-regular and the pairwise distance among the connected components in S is at least d for a prescribed degree r and a distance d. The problem includes a lot of famous graph optimization problems as special cases such as MATCHING, INDEPENDENT SET, INDUCED MATCHING, TWO FACTOR, HAMILTONIAN CYCLE, INDUCED LONGEST CYCLE problems. In this thesis we mainly consider two variants of the problem, MAXIMUM r-REGULAR INDUCED SUBGRAPH and DISTANCE-d INDEPENDENT SET problems, and we focus on the tractability / intractability, and the approximability / inapproximability of the problems on subclasses of graphs.

(1) First, we study the MAXIMUM r-REGULAR INDUCED SUBGRAPH problem, whose goal is to find a maximum vertex-subset S of an unweighted given graph G such that the subgraph $G[S]$ induced by S is r-regular for a prescribed degree $r \geq 0$. We also consider a variant of the problem which requires $G[S]$ to be r-regular and connected. Both problems are known to be NP-hard even to approximate for a fixed constant r. In this thesis, we thus consider the problems whose input graphs are restricted to some special classes of graphs. (i) We first show that the problems are still NP-hard to approximate even if r is a fixed constant and the input graph is either bipartite or planar. On the other hand, (ii) both problems are tractable for graphs having tree-like structures, as follows. We give linear-time algorithms to solve the problems for graphs with bounded treewidth; we note that the hidden constant factor of our running time is just a single exponential of the treewidth. Furthermore, (iii) both problems are solvable in polynomial time for chordal graphs.

(2) Next, we study the DISTANCE-d INDEPENDENT SET problem, which is a generalization of the INDEPENDENT SET problem (IS for short). A distance-d independent set for an integer $d \geq 2$ in an unweighted graph $G = (V, E)$ is a subset $S \subseteq V$ of vertices such that for any pair of vertices $u, v \in S$, the distance between u and v is at least d in G. Given an unweighted graph G and a positive integer k, the DISTANCE-d INDEPENDENT SET problem (DdIS for short) is to decide

whether G contains a distance-d independent set S such that $|S| \geq k$. D2IS is identical to the original IS. Thus D2IS is $\mathcal{NP}$-complete even for planar graphs, but it is in $\mathcal{P}$ for bipartite graphs and chordal graphs. In this thesis we investigate the computational complexity of DdIS, its maximization version MaxDdIS, and its parameterized version ParaDdIS(k), where the parameter is the size of the distance-d independent set: (i) We first prove that for any $\varepsilon > 0$ and any fixed integer $d \geq 3$, it is $\mathcal{NP}$-hard to approximate MaxDdIS to within a factor of $n^{1/2-\varepsilon}$ for bipartite graphs of n vertices, and for any fixed integer $d \geq 3$, ParaDdIS(k) is $\mathcal{W}[1]$-hard for bipartite graphs. Then, (ii) we prove that for every fixed integer $d \geq 3$, DdIS remains $\mathcal{NP}$-complete even for planar bipartite graphs of maximum degree three. Furthermore, (iii) we show that if the input graph is restricted to chordal graphs, then DdIS can be solved in polynomial time for any even $d \geq 2$, whereas DdIS is $\mathcal{NP}$-complete for any odd $d \geq 3$. Also, we show the hardness of approximation of MaxDdIS and the $\mathcal{W}[1]$-hardness of ParaDdIS(k) on chordal graphs for any odd $d \geq 3$.

Acknowledgments

First, and foremost, I am deeply indebted to my academic advisor, Professor Miyano, for being enthusiastic supervisor. I am grateful to Eiji for his perceptiveness, and his deep insights over the year on my research. He provided me a great source of inspiration. Eiji have been extremely generous in giving me a great deal of his valuable time and sharing with me his ideas and insights. This research would have been impossible without his help.

I thank Professor Yuichi Asahiro and Professor Takehiro Ito for fruitful collaborations and discussions in the areas related to this thesis. Furthermore, I thank Professor Katsuhiro Inoue, Professor Eitaku Nobuyama, and Professor Akihiro Fujiwara for many valuable comments.

CONTENTS

Chapter 1

Introduction

This thesis deals with the problem of finding a maximum vertex-subset S of a given a graph such that the subgraph induced by S is r-regular and the pairwise distance among the connected components in S is at least d for a prescribed degree r and a distance d. The problem includes a lot of famous graph optimization problems as special cases such as MATCHING, INDEPENDENT SET, INDUCED MATCHING, TWO FACTOR, HAMILTONIAN CYCLE, INDUCED LONGEST CYCLE problems. In this thesis we mainly consider two variants of the problem, MAXIMUM r-REGULAR INDUCED SUBGRAPH and DISTANCE-d INDEPENDENT SET problems, and we focus on the tractability / intractability, and the approximability / inapproximability of the problems on subclasses of graphs.

The problem MAXIMUM INDUCED SUBGRAPH for a fixed property Π is the following class of problems [17, GT21]: Given a graph G, find a maximum vertex-subset such that its induced subgraph of G satisfies the property Π. The problem MAXIMUM INDUCED SUBGRAPH is very universal; a lot of graph optimization problems can be formulated as MAXIMUM INDUCED SUBGRAPH by specifying the property Π appropriately. For example, if the property Π is "bipartite," then we wish to find the largest induced bipartite subgraph of a given graph G. Therefore, MAXIMUM INDUCED SUBGRAPH is one of the most important problems in the fields of graph theory and combinatorial optimization, and thus has been extensively studied over the past few decades. Unfortunately, however, it has been shown that MAXIMUM INDUCED SUBGRAPH is intractable for a large class of interesting properties. For example, Lund and Yannakakis [30] proved that MAXIMUM INDUCED SUBGRAPH for natural properties, such as planar, outerplanar, bipartite, complete bipartite, acyclic, degree-constrained, chordal and interval, are all NP-hard even to approximate.

Furthermore, one of the most important problems is the INDEPENDENT SET problem, i.e., the property Π is "induced subgraph is an independent set." The input of INDEPENDENT SET is an unweighted graph $G = (V, E)$ and a positive integer $k \leq |V|$. An *independent set* of G is a subset $S \subseteq V$ of vertices such that, for all $u, v \in S$, the edge $\{u, v\}$ is not in E. INDEPENDENT SET asks whether G contains an independent set S having $|S| \geq k$. INDEPENDENT SET is among the first problems ever to be shown to be $\mathcal{NP}$-complete, and has been used as a starting point for proving the $\mathcal{NP}$-completeness of other problems [17]. Moreover, it is well known that INDEPENDENT SET remains $\mathcal{NP}$-complete even for substantially restricted graph classes such as cubic planar graphs [16], triangle-free graphs [34], and graphs with large girth [32].

In MAXIMUM INDUCED SUBGRAPH problem, the distance of each vertices in each induced subgraph is at least 2. Furthermore, the problems when $r = 0$ correspond to the well studied MAXIMUM INDEPENDENT SET problems. In this thesis, we consider a generalization of MAXIMUM INDEPENDENT SET, named the MAXIMUM r-REGULAR INDUCED SUBGRAPH problem and the DISTANCE-d INDEPENDENT SET problem. MAXIMUM r-REGULAR INDUCED SUBGRAPH problem for an integer $r \geq$ is r-regular and distance at least 2. Furthermore, DISTANCE-d INDEPENDENT SET problem for an integer $d \geq 2$ is 0-regular and distance at least d.

In Chapter 3, we consider MAXIMUM r-REGULAR INDUCED SUBGRAPH problem.

MAXIMUM r-REGULAR INDUCED SUBGRAPH (r-MaxRIS)	
Input:	A graph $G = (V, E)$.
Goal:	Find a maximum vertex-subset $S \subseteq V$ such that the subgraph induced by S is r-regular.

The optimal value (*i.e.*, the number of vertices in an optimal solution) to r-MaxRIS for a graph G is denoted by $\mathrm{OPT}_{\mathsf{RIS}}(G)$. Consider, for example, the graph G in Fig. 3.7(a) as an input of 3-MaxRIS. Then, the three connected components induced by the white vertices have the maximum size of 12, that is, $\mathrm{OPT}_{\mathsf{RIS}}(G) = 12$. Notice that r-MaxRIS for $r = 0$ and $r = 1$ correspond to the well-studied problems MAXIMUM INDEPENDENT SET [17, GT20] and MAXIMUM INDUCED MATCHING [9], respectively.

(a) (b)

Fig. 1.1 Optimal solutions for (a) 3-MaxRIS and (b) 3-MaxRICS.

We also study the following variant which requires the *connectivity* property in addition to the regularity property. (This variant can be seen as the special case of the problem MAXIMUM INDUCED CONNECTED SUBGRAPH for a fixed property Π [17, GT22].)

MAXIMUM r-REGULAR INDUCED CONNECTED SUBGRAPH (r-MaxRICS)	
Input:	A graph $G = (V, E)$.
Goal:	Find a maximum vertex-subset $S \subseteq V$ such that the subgraph induced by S is r-regular and connected.

The optimal value to r-MaxRICS for a graph G is denoted by $\mathrm{OPT}_{\mathsf{RICS}}(G)$. For the graph G in Fig. 3.7(b), which is the same as one in Fig. 3.7(a), the subgraph induced by the white vertices has the maximum size of six for 3-MaxRICS, that is, $\mathrm{OPT}_{\mathsf{RICS}}(G) = 6$. Notice that r-MaxRICS for $r = 0, 1$ is trivial for any graph; it simply finds one vertex for $r = 0$, and one edge for $r = 1$. On the other hand, 2-MaxRICS is known as the LONGEST INDUCED CYCLE problem which is NP-hard [17, GT23].

We prove that the inapproximability result of $n^{1/6-\varepsilon}$ in the case $r \geq 3$ can be improved to

$n^{1/2-\varepsilon}$.Also, we show the following parameterized complexity of r-ParaRICS by use similar with a small mofdification from the gap-preserving reduction to an *fpt reduction*.

Furthermore, we study the problems r-MaxRIS and r-MaxRICS from the viewpoint of graph classes: Are they tractable if input graphs have special structures? We first show that r-MaxRIS and r-MaxRICS are NP-hard to approximate even if the input graph is either bipartite or planar. Then, we consider the problems restricted to graphs having “tree-like” structures. More formally, we show that both r-MaxRIS and r-MaxRICS are solvable in linear time for graphs with bounded treewidth; we note that the hidden constant factor of our running time is just a single exponential of the treewidth. Furthermore, we show that the two problems are solvable in polynomial time for chordal graphs. The formal definitions of these graph classes will be given later, but it is important to note that they have the following relationships (see, e.g., [8]): (1) there is a planar graph with n vertices whose treewidth is $\Omega(\sqrt{n})$; and (2) both chordal and bipartite graphs are well-known subclasses of perfect graphs. As a brief summary, our results show that both problems are still intractable for graphs with treewidth $\Omega(\sqrt{n})$, while they are tractable if the treewidth is bounded by a fixed constant. Since our problems are intractable for bipartite graphs, they are intractable for perfect graphs, too; but the “chordality” makes the problems tractable.

It is known that any optimization problem that can be expressed by Extended Monadic Second Order Logic (EMSOL) can be solved in linear time for graphs with bounded treewidth [11]. However, the algorithm obtained by this method is hard to implement, and is very slow since the hidden constant factor of the running time is a tower of exponentials of unbounded height with respect to the treewidth [28]. On the other hand, our algorithms are simple, and the hidden constant factor is just a single exponential of the treewidth.

Our main results are summarized in the following list:

(i) Let $\rho(n) \geq 1$ be any polynomial-time computable function. For every fixed integer $r \geq 3$ and bipartite graphs of maximum degree $r+1$, r-MaxRIS and r-MaxRICS admit no polynomial-time approximation algorithm within a factor of $\rho(n)$ unless P = NP.

(ii) For every fixed constant $r \geq 0$, r-MaxRIS is solvable in linear time for graphs with bounded treewidth.

(iii) For every fixed constant $r \geq 0$, r-MaxRICS is solvable in linear time for graphs with bounded treewidth.

(iv) Let $\rho(n) \geq 1$ be any polynomial-time computable function. For every fixed integer r, $3 \leq r \leq 5$, r-MaxRIS and r-MaxRICS for planar graphs admit no polynomial-time approximation algorithm within a factor of $\rho(n)$ unless P = NP.

(v) For every integer $r \geq 0$, r-MaxRIS can be solved in time $O(n^2)$ for chordal graphs, where n is the number of vertices in a given graph.

(vi) For every integer $r \geq 0$, r-MaxRICS is solvable in polynomial time for chordal graphs.

In Chapter 4, we consider a generalization of IS, named the Distance-d Independent Set problem (DdIS for short). A distance-d independent set for an integer $d \geq 2$ in an unweighted graph $G = (V, E)$ is a subset $S \subseteq V$ of vertices such that for any pair of vertices $u, v \in S$, the distance between u and v is at least d in G. For a fixed constant $d \geq 2$, DdIS considered in this thesis is formulated as the following class of problems [1]:

Distance-d Independent Set (DdIS)

Input: An unweighted graph $G = (V, E)$ and a positive integer $k \leq |V|$.

Question: Does G contain a distance-d independent set of size k or more?

The maximization version of DdIS can be also defined:

MAXIMUM DISTANCE-d INDEPENDENT SET (MaxDdIS)
Input: An unweighted graph $G = (V, E)$.
Output: A distance-d independent set of the maximum size.

The problem parameterized by the solution size k is as follows:

PARAMETERIZED DISTANCE-d INDEPENDENT SET (ParaDdIS(k))
Input: An unweighted graph $G = (V, E)$.
Parameter A positive integer $k \leq \|V\|$.
Question: Does G contain a distance-d independent set of size k or more?

It is important to note that D2IS is identical to the original IS, and DdIS is equivalent to IS on the $(d-1)$th power graph G^{d-1} of the input graph G as pointed out in [1].

Even when $d = 2$, DdIS (i.e., D2IS) is $\mathcal{NP}$-complete, and thus it would be easy to show that DdIS is $\mathcal{NP}$-complete in general. Fortunately, however, it is known that if the input graph is restricted to, for example, bipartite graphs [22], chordal graphs [18], circular-arc graphs [19], comparability graphs [20], and many other classes [31, 29, 7], then D2IS admits polynomial-time algorithms. Furthermore, Agnarsson, Damaschke, Halldórsson [1] show the following tractability of DdIS by using the closure property under taking power [14, 15, 35]:

Fact 1 ([1]) *Let n denote the number of vertices in the input graph G. Then, for every integer $d \geq 2$, DdIS is solvable in $O(n)$ time for interval graphs, in $O(n(\log\log n + \log d))$ time for trapezoid graphs, and in $O(n)$ time for circular-arc graphs.*

This tractability suggests that if we restrict the set of instances to, for example, subclasses of bipartite graphs and chordal graphs, then DdIS for a fixed $d \geq 3$ might be also solvable efficiently. On the other hand, however, we have a "negative" fact that if G is planar/bipartite, then the $(d-1)$th power graph G^{d-1} is not necessarily planar/bipartite. From those points of view, this thesis investigates DdIS, namely, our work focuses on the computational complexity of DdIS and/or the inapproximability of MaxDdIS on (subclasses of) bipartite graphs and chordal graphs.

Our main results are summarized in the following list:

(i) For every fixed integer $d \geq 3$, DdIS is $\mathcal{NP}$-complete even for bipartite graphs.

(ii) For any $\varepsilon > 0$ and fixed integer $d \geq 3$, it is $\mathcal{NP}$-hard to approximate MaxDdIS to within a factor of $n^{1/2-\varepsilon}$ for bipartite graphs of n vertices.

(iii) For every fixed integer $d \geq 3$, ParaDdIS(k) is $\mathcal{W}[1]$-hard for bipartite graphs.

(iv) For every fixed integer $d \geq 3$, DdIS remains $\mathcal{NP}$-complete even for planar bipartite graphs of maximum degree three.

(v) For every fixed even integer $d \geq 2$, DdIS is in $\mathcal{P}$ for chordal graphs.

(vi) For every fixed odd integer $d \geq 3$, DdIS is $\mathcal{NP}$-complete for chordal graphs.

(vii) For any $\varepsilon > 0$ and fixed odd integer $d \geq 3$, it is $\mathcal{NP}$-hard to approximate MaxDdIS to within a factor of $n^{1/2-\varepsilon}$ for chordal graphs of n vertices.

(viii) For every fixed odd integer $d \geq 3$, ParaDdIS(k) is $\mathcal{W}[1]$-hard for chordal graphs.

Chapter 2

Preliminaries

For the maximization problems, an algorithm `ALG` is called a σ-approximation algorithm and the approximation ratio of `ALG` is σ if $OPT(G)/ALG(G) \leq \sigma$ holds for every input G, where $ALG(G)$ and $OPT(G)$ are the number of vertices of obtained subsets by `ALG` and the number of vertices of an optimal solution, respectively.

Let MaxP_1 and MaxP_2 be maximization problems. A *gap-preserving reduction*, say, Γ, from MaxP_1 to MaxP_2 comes with four parameter functions, g_1, α, g_2 and β. Given an instance x of MaxP_1, the reduction Γ computes an instance y of MaxP_2 in polynomial time such that if $OPT_{\mathsf{MaxP}_1}(x) \geq g_1(x)$, then $OPT_{\mathsf{MaxP}_2}(y) \geq g_2(y)$, and if $OPT_{\mathsf{MaxP}_1}(x) < g_1(x)/\alpha(|x|)$, then $OPT_{\mathsf{MaxP}_2}(y) < g_2(y)/\beta(|y|)$, where $OPT_{\mathsf{MaxP}_1}(x)$ and $OPT_{\mathsf{MaxP}_2}(y)$ denote the objective function values of optimal solutions to the instances x and y, respectively. Note that $\alpha(|x|)$ is the approximation gap, i.e., the hardness factor of approximation for MaxP_1 and the gap-preserving reduction Γ shows that there is no $\beta(|y|)$ factor approximation algorithm for MaxP_2 unless $\mathcal{P} = \mathcal{NP}$ (see, e.g., Chapter 29 in [36]).

A *parameterized problem* is a pair (Q, k) where $Q \subseteq \Sigma^*$ is a decision problem over some alphabet Σ, and $k : \Sigma^* \to \mathbf{N}$ is a *parameterization* of the problem, assigning a *parameter* to each instance of Q. An algorithm is *fixed-parameter tractable* or *fpt* if it has a running time at most $f(k) \cdot n^c$ for some computable function f and a constant c, where n is the input length and k is the parameter assigned to the input. Given two parameterized problems (Q_1, k_1) and (Q_2, k_2) over the alphabet Σ, an *fpt-reduction* from (Q_1, k_1) to (Q_2, k_2) is a function $g : \Sigma^* \to \Sigma^*$, computable by an fpt-algorithm, such that $I \in Q_1$ if and only if $g(I) \in Q_2$ and $k_2(g(I)) \leq f(k_1(I))$ for some computable function f, for every $I \in \Sigma^*$.

Now, we define graph theoretic notations that need in this thesis. The definitions of graph classes are from [8] and [21]:

Regular. A graph is *r-regular* if the degree of every vertex is exactly r.

Distance. For a pair of vertices u and v, the length of a shortest path from u to v, i.e., the distance between u and v is denoted by $dist_G(u, v)$, and the diameter G is defined as $diam(G) = \max_{u,v \in V} dist_G(u, v)$.

Planar Graph. An undirected graph G is *planar* if and only if it does not contain a subdivision of K_5 or $K_{3,3}$.

Bipartite graph. An undirected graph $G = (V, E)$ is *bipartite* if its vertices can be partitioned into two disjoint independent sets $V = S_1 + S_2$, i.e., every edge has one endpoint in S_1 and the other in S_2.

Split graph. An undirected graph $G = (V, E)$ is defined to be *split* if there is a partition $V = S + K$ of its vertex set into an independent set S and complete set K. There is no restriction on edges between vertices of S and vertices of K. In general, the partition $V = S + K$ of split graph will not be unique; neither will S(resp.K) necessarily be a maximal independent set(resp.clique).

Chrdal graph and Cliquetree. A graph G is *chordal* if every cycle in G of length at least four has at least one chord, which is an edge joining non-adjacent vertices in the cycle [8]. (See Fig. 3.19(a) as an example.)

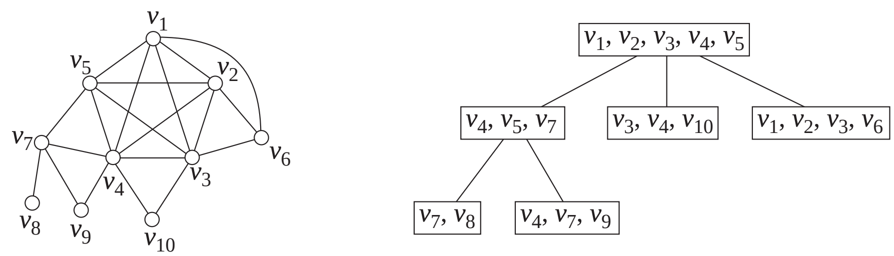


Fig. 2.1 (a) Chordal graph G and (b) its clique tree T.

Let $\mathcal{K}_G$ be the set of all maximal cliques in a graph G, and let $\mathcal{K}_v \subseteq \mathcal{K}_G$ be the set of all maximal cliques that contain a vertex $v \in V(G)$. It is known that G is chordal if and only if there exists a tree $T = (\mathcal{K}_G, E)$ such that each node of T corresponds to a maximal clique in $\mathcal{K}_G$ and the induced subtree $T[\mathcal{K}_v]$ is connected for every vertex $v \in V(G)$ [5]. (See Fig. 3.19 as an example.) Such a tree is called a *clique tree* of G, and it can be constructed in linear time [5]. Indeed, a clique tree of a chordal graph G is a tree-decomposition of G.

Treewidth and Tree-decomposition. Let G be a graph with n vertices. A *tree-decomposition* of G is a pair $\langle \{X_i \mid i \in V_T\}, T \rangle$, where $T = (V_T, E_T)$ is a rooted tree, such that the following four conditions (1)–(4) hold [6]:

(1) each X_i is a subset of $V(G)$, and is called a *bag*;

(2) $\bigcup_{i \in V_T} X_i = V(G)$;

(3) for each edge $(u, v) \in E(G)$, there is at least one node $i \in V_T$ such that $u, v \in X_i$; and

(4) for each vertex $v \in V(G)$, the set $\{i \in V_T \mid v \in X_i\}$ induces a connected component in T.

We will refer to a *node* in V_T in order to distinguish it from a vertex in $V(G)$. The *width* of a tree-decomposition $\langle \{X_i \mid i \in V_T\}, T \rangle$ is defined as $\max\{|X_i| - 1 : i \in V_T\}$, and the *treewidth* of G is the minimum k such that G has a tree-decomposition of width k.

In particular, a tree-decomposition $\langle \{X_i \mid i \in V_T\}, T \rangle$ of G is called a *nice tree-decomposition* if the following four conditions (5)–(8) hold [4]:

(5) $|V_T| = O(n)$;

(6) every node in V_T has at most two children in T;

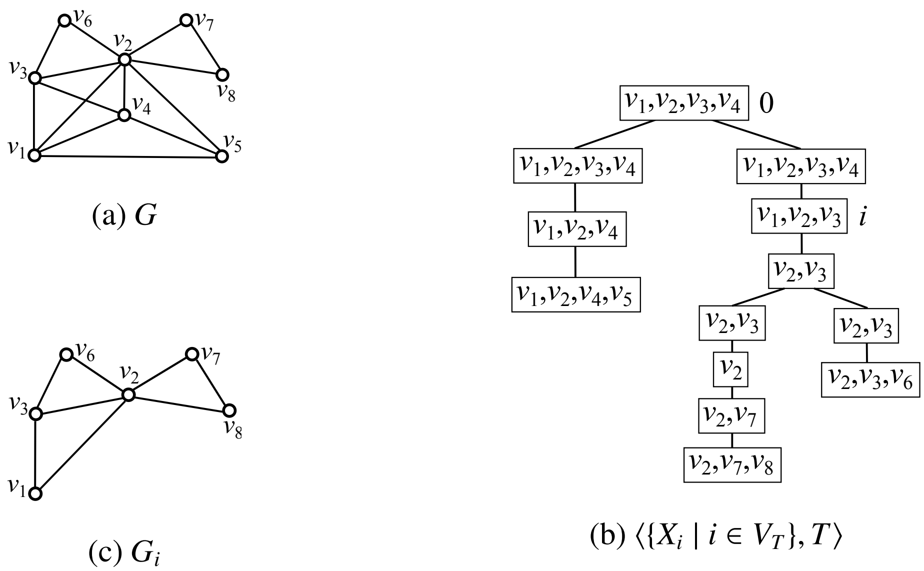
(7) if a node $i \in V_T$ has two children l and r, then $X_i = X_l = X_r$; and

(8) if a node $i \in V_T$ has only one child j, then one of the following two conditions (a) and (b) holds:

(a) $|X_i| = |X_j| - 1$ and $X_i \subset X_j$ (such a node i is called a *forget* node); and

(b) $|X_i| = |X_j| + 1$ and $X_i \supset X_j$ (such a node i is called an *introduce* node.)

Figure 2.2(b) illustrates a nice tree-decomposition $\langle\{X_i \mid i \in V_T\}, T\rangle$ of the graph G in Fig. 2.2(a) whose treewidth is three. It is known that any graph of treewidth k has a nice tree-decomposition of width k [4]. Since a nice tree-decomposition $\langle\{X_i \mid i \in V_T\}, T\rangle$ of a graph G with bounded treewidth can be found in linear time [4], we may assume without loss of generality that G and its nice tree-decomposition are both given.



(a) G

(b) $\langle\{X_i \mid i \in V_T\}, T\rangle$

(c) G_i

Fig. 2.2 (a) Graph G, (b) a nice tree-decomposition $\langle\{X_i \mid i \in V_T\}, T\rangle$ of G, and (c) the subgraph G_i of G for the node $i \in V_T$.

Each node $i \in V_T$ corresponds to a subgraph G_i of G which is induced by the vertices that are contained in the bag X_i and all bags of descendants of i in T. Therefore, if a node $i \in V_T$ has two children l and r in T, then G_i is the union of G_l and G_r which are the subgraphs corresponding to nodes l and r, respectively. Clearly, $G = G_0$ for the root 0 of T. For example, Fig. 2.2(c) illustrates the subgraph G_i of the graph G in Fig. 2.2(a) which corresponds to the node $i \in V_T$ in Fig. 2.2(b). By definitions (3) and (4) of a tree-decomposition, we have the following proposition.

Proposition 1 *For each node $i \in V_T$, there is no edge joining a vertex in $G_i \setminus X_i$ and one in $G \setminus G_i$.*

Chapter 3

Regular Induced Subgraphs

3.1 Introducion

Recall from Chapter 1 that the problem Maximum Induced Subgraph (MaxIS) for a fixed property Π is the following class of problems [17, GT21]: Given a graph G, find a maximum vertex-subset such that its induced subgraph of G satisfies the property Π. The problem MaxIS is very universal; a lot of graph optimization problems can be formulated as MaxIS by specifying the property Π appropriately. For example, if the property Π is "bipartite," then we wish to find the largest induced bipartite subgraph of a given graph G. Therefore, MaxIS is one of the most important problems in the fields of graph theory and combinatorial optimization, and thus has been extensively studied over the past few decades. Unfortunately, however, it has been shown that MaxIS is intractable for a large class of interesting properties. For example, Lund and Yannakakis [30] proved that MaxIS for natural properties, such as planar, outerplanar, bipartite, complete bipartite, acyclic, degree-constrained, chordal and interval, are all NP-hard even to approximate.

(a) (b)

Fig. 3.1 Optimal solutions for (a) 3-MaxRIS and (b) 3-MaxRICS.

3.1.1 Our problems

In this chapter, we consider another natural and fundamental property, that is, the *regularity* of graphs. A graph is *r-regular* if the degree of every vertex is exactly $r \geq 0$. We study the following variant of MaxIS:

MAXIMUM r-REGULAR INDUCED SUBGRAPH (r-MaxRIS)	
Input:	A graph $G = (V, E)$.
Goal:	Find a maximum vertex-subset $S \subseteq V$ such that the subgraph induced by S is r-regular.

The optimal value (*i.e.*, the number of vertices in an optimal solution) to r-MaxRIS for a graph G is denoted by $\mathrm{OPT}_{\mathsf{RIS}}(G)$. Consider, for example, the graph G in Fig. 3.7(a) as an input of 3-MaxRIS. Then, the three connected components induced by the white vertices have the maximum size of 12, that is, $\mathrm{OPT}_{\mathsf{RIS}}(G) = 12$. Notice that r-MaxRIS for $r = 0$ and $r = 1$ correspond to the well-studied problems MAXIMUM INDEPENDENT SET [17, GT20] and MAXIMUM INDUCED MATCHING [9], respectively.

We also study the following variant which requires the *connectivity* property in addition to the regularity property. (This variant can be seen as the special case of the problem MAXIMUM INDUCED CONNECTED SUBGRAPH for a fixed property Π [17, GT22].)

MAXIMUM r-REGULAR INDUCED CONNECTED SUBGRAPH (r-MaxRICS)	
Input:	A graph $G = (V, E)$.
Goal:	Find a maximum vertex-subset $S \subseteq V$ such that the subgraph induced by S is r-regular and connected.

The optimal value to r-MaxRICS for a graph G is denoted by $\mathrm{OPT}_{\mathsf{RICS}}(G)$. For the graph G in Fig. 3.7(b), which is the same as one in Fig. 3.7(a), the subgraph induced by the white vertices has the maximum size of six for 3-MaxRICS, that is, $\mathrm{OPT}_{\mathsf{RICS}}(G) = 6$. Notice that r-MaxRICS for $r = 0, 1$ is trivial for any graph; it simply finds one vertex for $r = 0$, and one edge for $r = 1$. On the other hand, 2-MaxRICS is known as the LONGEST INDUCED CYCLE problem which is NP-hard [17, GT23].

Furthermore, we consider the parameterized variant of r-MaxRICS:

PARAMETERIZED r-REGULAR INDUCED CONNECTED SUBGRAPH (r-ParaRICS)	
Input:	A graph $G = (V, E)$ and an integer k.
Parameter:	k
Problem:	Decide whether there is a subset of vertices $S \subseteq V$ with $\lvert S \rvert \geq k$ such that the induced subgraph $G[S]$ on S is connected and r-regular.

3.1.2 Related Work

Both r-MaxRIS and r-MaxRICS include a variety of well-known problems, and hence they have been widely studied in the literature. Below, let n be the number of vertices in a given graph and assume that $\mathrm{P} \neq \mathrm{NP}$.

For r-MaxRIS, as mentioned above, two of the most well-studied and important problems must be MAXIMUM INDEPENDENT SET (*i.e.*, 0-MaxRIS) and MAXIMUM INDUCED MATCHING (*i.e.*, 1-MaxRIS). Unfortunately, however, they are NP-hard even to approximate. Håstad [23] proved that 0-MaxRIS cannot be approximated in polynomial time within a factor of $n^{1/2-\varepsilon}$ for any $\varepsilon > 0$. Orlovich, Finke, Gordon and Zverovich [33] showed the inapproximability of a factor of $n^{1/2-\varepsilon}$ for 1-MaxRIS for any $\varepsilon > 0$. Moreover, for any fixed integer $r \geq 3$, Cardoso, Kamiński and Lozin [10] proved that r-MaxRIS is NP-hard.

For r-MaxRICS, that is, the variant with the connectivity property, Kann [27] proved that LONGEST INDUCED CYCLE (*i.e.*, 2-MaxRICS) cannot be approximated within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$.

A related problem is finding a maximum subgraph which satisfies the regularity property but is not necessarily an induced subgraph of a given graph. This problem has been also studied extensively: for example, it is known to be NP-complete to determine whether there exists a 3-regular subgraph in a given graph [17, GT32]. Furthermore, Stewart proved that it remains NP-complete even if the input graph is either planar [24, 25] or bipartite [26].

3.1.3 Contributions

First we study the problem$r$-MaxRICS in general graphs. we prove that the inapproximability result of $n^{1/6-\varepsilon}$ in the case $r \geq 3$ can be improved to $n^{1/2-\varepsilon}$.Also, we show the following parameterized complexity of r-ParaRICS by use similar with a small mofdification from the gap-preserving reduction to an *fpt reduction*.

Furthermore, we study the problems r-MaxRIS and r-MaxRICS from the viewpoint of graph classes: Are they tractable if input graphs have special structures? We first show that r-MaxRIS and r-MaxRICS are NP-hard to approximate even if the input graph is either bipartite or planar. Then, we consider the problems restricted to graphs having "tree-like" structures. More formally, we show that both r-MaxRIS and r-MaxRICS are solvable in linear time for graphs with bounded treewidth; we note that the hidden constant factor of our running time is just a single exponential of the treewidth. Furthermore, we show that the two problems are solvable in polynomial time for chordal graphs. The formal definitions of these graph classes will be given later, but it is important to note that they have the following relationships (see, e.g., [8]): (1) there is a planar graph with n vertices whose treewidth is $\Omega(\sqrt{n})$; and (2) both chordal and bipartite graphs are well-known subclasses of perfect graphs. As a brief summary, our results show that both problems are still intractable for graphs with treewidth $\Omega(\sqrt{n})$, while they are tractable if the treewidth is bounded by a fixed constant. Since our problems are intractable for bipartite graphs, they are intractable for perfect graphs, too; but the "chordality" makes the problems tractable.

It is known that any optimization problem that can be expressed by Extended Monadic Second Order Logic (EMSOL) can be solved in linear time for graphs with bounded treewidth [11]. However, the algorithm obtained by this method is hard to implement, and is very slow since the hidden constant factor of the running time is a tower of exponentials of unbounded height with respect to the treewidth [28]. On the other hand, our algorithms are simple, and the hidden constant factor is just a single exponential of the treewidth.

Our main results are summarized in the following list:

(i) Let $\rho(n) \geq 1$ be any polynomial-time computable function. For every fixed integer $r \geq 3$ and bipartite graphs of maximum degree $r+1$, r-MaxRIS and r-MaxRICS admit no polynomial-time approximation algorithm within a factor of $\rho(n)$ unless P = NP.

(ii) For every fixed constant $r \geq 0$, r-MaxRIS is solvable in linear time for graphs with bounded treewidth.

(iii) For every fixed constant $r \geq 0$, r-MaxRICS is solvable in linear time for graphs with bounded treewidth.

(iv) Let $\rho(n) \geq 1$ be any polynomial-time computable function. For every fixed integer r, $3 \leq r \leq 5$, r-MaxRIS and r-MaxRICS for planar graphs admit no polynomial-time approximation algorithm within a factor of $\rho(n)$ unless P = NP.

(v) For every integer $r \geq 0$, r-MaxRIS can be solved in time $O(n^2)$ for chordal graphs, where n is the number of vertices in a given graph.

(vi) For every integer $r \geq 0$, r-MaxRICS is solvable in polynomial time for chordal graphs.

3.1.4 Notation

Let $G = (V, E)$ be a graph; we sometimes denote by $V(G)$ and $E(G)$ the vertex set and edge set of G, respectively. By (u, v) we denote an edge with endpoints u and v. For a vertex u, the set of vertices adjacent to u in G is denoted by $N(G, u)$ or simply by $N(u)$, and $(u, N(u))$ denotes the set $\{(u, v) \mid v \in N(u)\}$ of edges. Let the degree of a vertex u be denoted by $d(G, u)$ or $deg(u)$, i.e., $deg(u) = |N(u)|$. For a subset V' of $V(G)$, we denote by $G[V']$ the subgraph of G induced by V'; recall that a subgraph of G is said to be *induced* by V' if it contains all edges in $E(G)$ whose endpoints are both in V'. We denote simply by $G \setminus V'$ the induced subgraph $G[V \setminus V']$. For a subgraph G' of G, let $G \setminus G' = G \setminus V(G')$.

A (simple) path P of length ℓ from a vertex v_0 to a vertex v_ℓ is represented as a sequence of vertices such that $P = \langle v_0, v_1, \cdots, v_\ell \rangle$, and $|P|$ denotes the length of P. A cycle C of length ℓ is similarly denoted by $C = \langle v_0, v_1, \cdots, v_{\ell-1}, v_0 \rangle$, and $|C|$ denotes the length of C. A *chord* of a path (cycle) $\langle v_0, \cdots, v_\ell \rangle$ ($\langle v_0, \cdots, v_{\ell-1}, v_0 \rangle$) is an edge between two vertices of the path (cycle) that is not an edge of the path (cycle). A path (cycle) is *chordless* if it contains no chords, i.e., an induced cycle must be chordless. Let G_1, G_2, $\cdots$, G_ℓ be ℓ graphs and also let v_i and v'_i be two vertices in G_i for $1 \leq i \leq \ell$. Then, $\langle G_1, G_2, \cdots, G_\ell \rangle$ denotes a graph $G = (V(G_1) \cup V(G_2) \cup \cdots \cup V(G_\ell), E(G_1) \cup E(G_2) \cup \cdots \cup E(G_\ell) \cup \{(v'_1, v_2), (v'_2, v_3), \cdots, (v'_{\ell-1}, v_\ell)\})$. That is, two adjacent graphs G_{i-1} and G_i are connected by only one edge (v'_{i-1}, v_i) and G roughly forms a path, which will be called *path-like structure*. Similarly, $\langle G_1, G_2, \cdots, G_\ell, G_1 \rangle$ roughly forms a cycle, which will be called *cycle-like structure*.

Let MaxP_1 and MaxP_2 be maximization problems. A *gap-preserving reduction*, say, Γ, from MaxP_1 to MaxP_2 comes with four parameter functions, g_1, α, g_2 and β. Given an instance x of MaxP_1, the reduction Γ computes an instance y of MaxP_2 in polynomial time such that if $OPT_{\mathsf{MaxP}_1}(x) \geq g_1(x)$, then $OPT_{\mathsf{MaxP}_2}(y) \geq g_2(y)$, and if $OPT_{\mathsf{MaxP}_1}(x) < g_1(x)/\alpha(|x|)$, then $OPT_{\mathsf{MaxP}_2}(y) < g_2(y)/\beta(|y|)$, where $OPT_{\mathsf{MaxP}_1}(x)$ and $OPT_{\mathsf{MaxP}_2}(y)$ denote the objective function values of optimal solutions to the instances x and y, respectively. Note that $\alpha(|x|)$ is the approximation gap, i.e., the hardness factor of approximation for MaxP_1 and the gap-preserving reduction Γ shows that there is no $\beta(|y|)$ factor approximation algorithm for MaxP_2 unless $\mathcal{P} = \mathcal{NP}$ (see, e.g., Chapter 29 in [36]).

3.2 Hardness of Approximating r-MaxRICS

In this section we give the proofs of Theorem 1 and Corollary 1. The hardness of approximating r-MaxRICS for $r \geq 3$ is shown via a gap-preserving reduction from LONGEST INDUCED CYCLE problem, i.e., 2-MaxRICS. Consider an input graph $G = (V(G), E(G))$ of 2-MaxRICS with n vertices and m edges. Then, we construct a graph $H = (V(H), E(H))$ of r-MaxRICS. First we show the $n^{1/6-\varepsilon}$ inapproximability of 3-MaxRICS and then the same $n^{1/6-\varepsilon}$ inapproximability of the general r-MaxRICS problem for $r \geq 4$.

Let $OPT_2(G)$ (and $OPT_r(H)$, respectively) denote the number of vertices of an optimal solution for G of 2-MaxRICS (and H of r-MaxRICS, respectively). Let $V(G) = \{v_1, v_2, \cdots, v_n\}$ of n vertices and $E(G) = \{e_1, e_2, \cdots, e_m\}$ of m edges. Let $g(n)$ be a parameter function of the instance G. Then we provide the gap preserving reduction such that (C1) if $OPT_2(G) \geq g(n)$, then $OPT_r(H) \geq 4(n^3 + 1) \times g(n)$, and (C2) if $OPT_2(G) < \frac{g(n)}{n^{1-\varepsilon}}$ for a positive constant ε, then $OPT_r(H) < 4(n^3 + 1) \times \frac{g(n)}{n^{1-\varepsilon}}$. As we will explain it, the number of vertices in the reduced graph H is $O(n^6)$. Hence the approximation gap is $n^{1-\varepsilon} = \Theta(|V(H)|^{1/6-\varepsilon})$ for any constant $\varepsilon > 0$. By redefining $|V(H)| = n$, we obtain the $n^{1/6-\varepsilon}$ inapproximability of r-MaxRICS.

Theorem 1 3-MaxRICS *cannot be approximated in polynomial time within a factor of* $n^{1/6-\varepsilon}$ *for any constant* $\varepsilon > 0$ *unless* $\mathcal{P} = \mathcal{NP}$*,where n is the number of vertices in the input graph.*

Furthermore, by using additional ideas to the reduction, we show the same inapproximability of r-MaxRICS for any fixed integer $r \geq 4$.

Corellary 1 *For any fixed integer* $r \geq 4$*,* r-MaxRICS *cannot be approximated in polynomial time within a factor of* $n^{1/6-\varepsilon}$ *for any constant* $\varepsilon > 0$ *unless* $\mathcal{P} = \mathcal{NP}$*,where n is the number of vertices in the input graph.*

The proofs of Theorem 1 and Corollary 1 will be given in Subsection 3.2.1 and 3.2.2.

3.2.1 Reduction for $r = 3$

Without loss of generality, we can assume that there is no vertex whose degree is one in the input graph G of 2-MaxRICS. The reason is that such a vertex does not contribute to any feasible solution, i.e., a cycle, of 2-MaxRICS and can be removed from G.

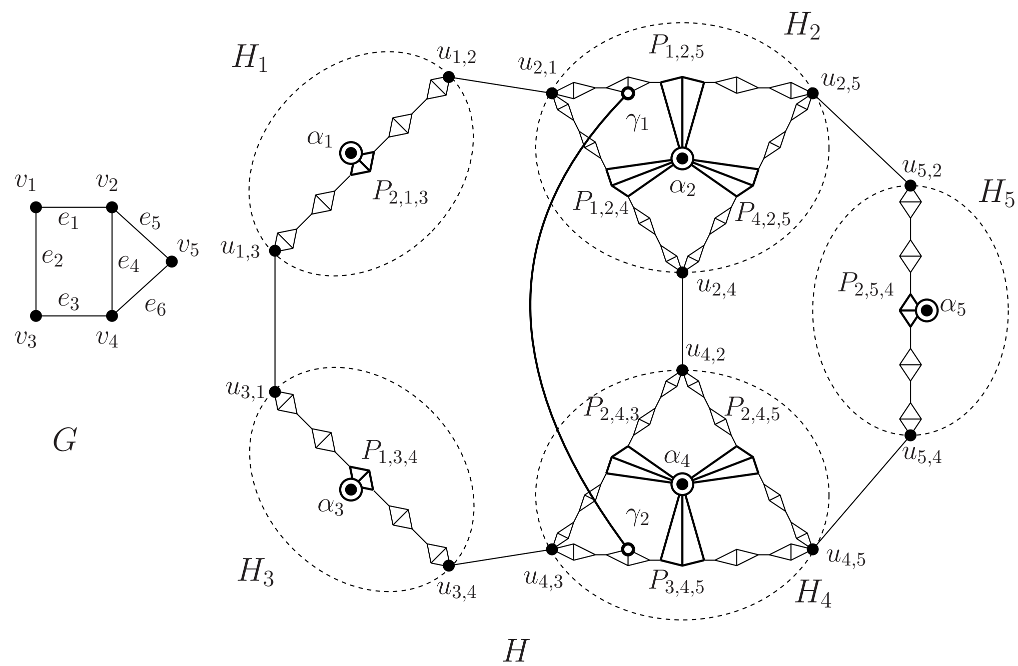


Fig. 3.2 Input graph G (left) and reduced graph H (right)

The constructed graph H consists of (i) n subgraphs, H_1 through H_n, which are associated with n vertices, v_1 through v_n, respectively, and (ii) m edge sets, E_1 through E_m, which are associated with m edges, e_1 through e_m, respectively. Now we only give a rough outline of the construction and explain the details later. See Figure 3.2. If an input instance G of 2-MaxRICS is the left graph, then the reduced graph H of 3-MaxRICS is illustrated in the right graph, where some details are omitted due to the space. Since the graph G has five vertices, v_1 through v_5, the graph H has five subgraphs, H_1 through H_5, each of which is illustrated by

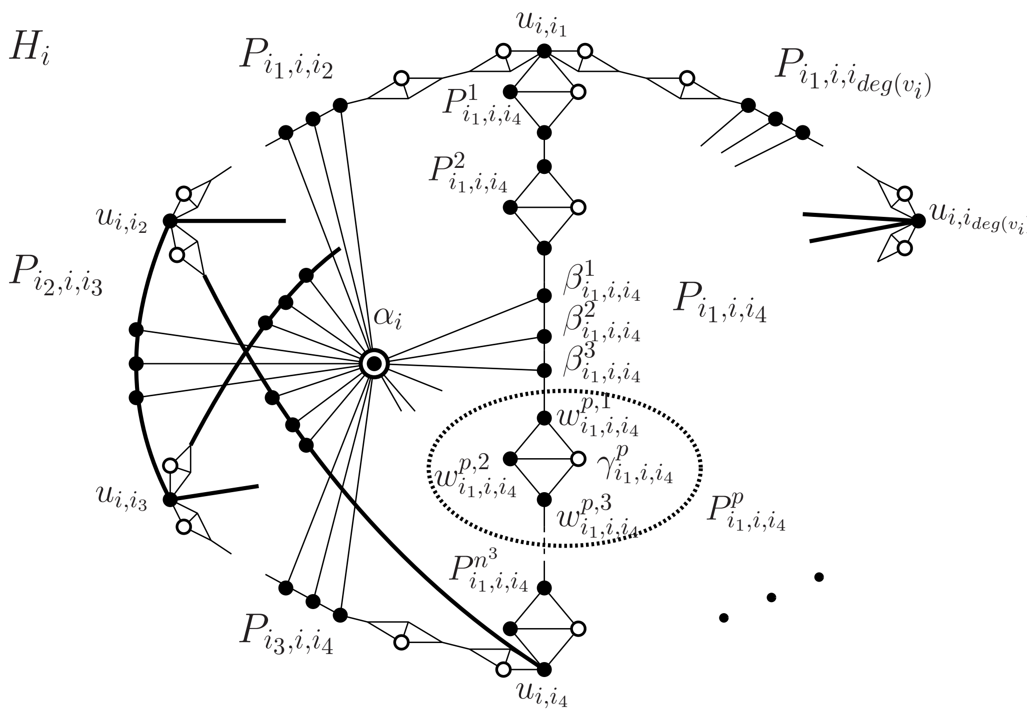


Fig. 3.3 Subgraph H_i

a dotted oval. One can see that each H_i roughly consists of $\binom{deg(v_i)}{2} = deg(v_i)(deg(v_i) - 1)/2$ path-like structures. For example, since two vertices v_1 and v_2 are connected via the edge e_1 in G, $u_{1,2}$ in H_1 is connected to $u_{2,1}$ in H_2. Similarly to e_2 through e_6, there are five edges, $(u_{1,3}, u_{3,1})$, $(u_{3,4}, u_{4,3})$, $(u_{2,4}, u_{4,2})$, $(u_{2,5}, u_{5,2})$, and $(u_{4,5}, u_{5,4})$ in H. The edge (γ_1, γ_2) between path-like structures labeled by $P_{1,2,5}$ in H_2 and by $P_{3,4,5}$ in H_4 plays an important role as described later.

(i) Here we describe the construction of the ith subgraph H_i in detail for every i ($1 \leq i \leq n$). See Figure 3.3, which illustrates H_i. Suppose that the set of vertices adjacent to v_i is $N(v_i) = \{v_{i_1}, v_{i_2}, \ldots, v_{i_{deg(v_i)}}\}$, where $i_j \in \{1, 2, \cdots n\} \setminus \{i\}$ for $1 \leq j \leq deg(v_i)$. The subgraph $H_i = (V(H_i), E(H_i))$ includes $deg(v_i)$ vertices, u_{i,i_1} through $u_{i,i_{deg(v_i)}}$ that correspond to the vertices adjacent to v_i, and $deg(v_i)(deg(v_i) - 1)/2$ *path gadgets*, P_{i_1,i,i_2}, P_{i_1,i,i_3}, $\cdots$, $P_{i_1,i,i_{deg(v_i)}}$, P_{i_2,i,i_3}, $\cdots$, $P_{i_{deg(v_i)-1},i,i_{deg(v_i)}}$, where two vertices u_{i,i_j} and u_{i,i_k} are connected via the path gadget

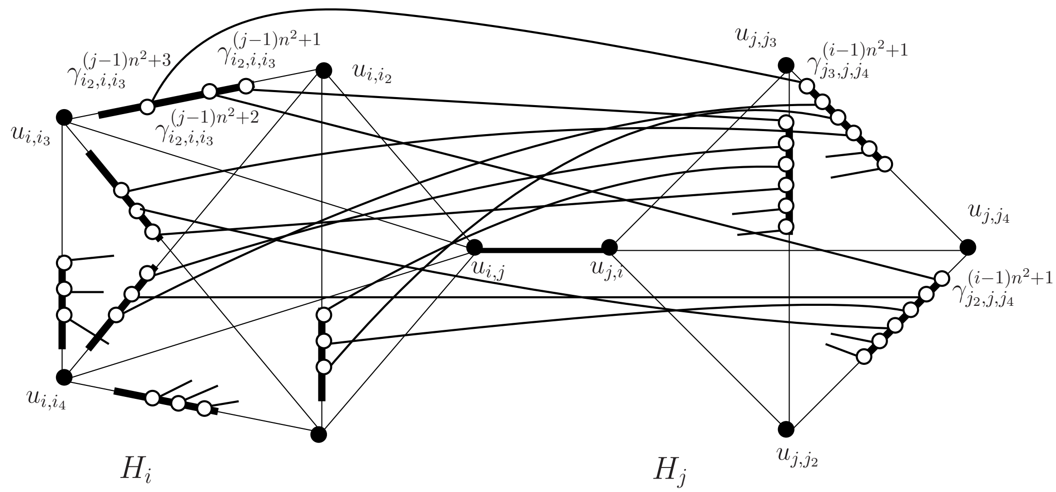


Fig. 3.4 E_k connecting H_i and H_j

P_{i_j,i,i_k} for $v_{i_j}, v_{i_k} \in N(v_i)$. As an example, in Figure 3.3, the top vertex u_{i,i_1} and the bottom u_{i,i_4} are connected via P_{i_1,i,i_4}. Each path gadget P_{i_j,i,i_k} includes n^3 subgraphs, $P^1_{i_j,i,i_k}$ through $P^{n^3}_{i_j,i,i_k}$, where, for each $1 \le p \le n^3$,

$$
\begin{aligned}
V(P^p_{i_j,i,i_k}) &= \{w^{p,1}_{i_j,i,i_k}, w^{p,2}_{i_j,i,i_k}, w^{p,3}_{i_j,i,i_k}, \gamma^p_{i_j,i,i_k}\},\\
E(P^p_{i_j,i,i_k}) &= (\gamma^p_{i_j,i,i_k}, \{w^{p,1}_{i_j,i,i_k}, w^{p,2}_{i_j,i,i_k}, w^{p,3}_{i_j,i,i_k}\})\\
&\quad \cup \{(w^{p,1}_{i_j,i,i_k}, w^{p,2}_{i_j,i,i_k}), (w^{p,2}_{i_j,i,i_k}, w^{p,3}_{i_j,i,i_k})\}.
\end{aligned}
$$

Note that the above number "n^3" of the subgraphs $P^p_{i_j,i,i_k}$'s comes from the upper bound of the total number of path gadgets: Each H_i contains $deg(v_i)(deg(v_i) - 1)/2$ path gadgets and thus, in total, $deg(v_i)(deg(v_i) - 1)/2 \times n$ path gadgets in H_1 through H_n, which is bounded above by n^3. Thus, we want to prepare n^3 subgraphs $P^p_{i_j,i,i_k}$'s (or, more precisely, we want to prepare n^3 γ-vertices which are defined later).

In the path gadget P_{i_j,i,i_k}, two vertices $w^{1,1}_{i_j,i,i_k}$ and $w^{n^3,3}_{i_j,i,i_k}$ are respectively identical to the vertices u_{i,i_j} and u_{i,i_k} prepared in the above. For $2 \le p \le n^3$, contiguous two subgraphs $P^{p-1}_{i_j,i,i_k}$ and $P^p_{i_j,i,i_k}$ are connected by one edge $(w^{p-1,3}_{i_j,i,i_k}, w^{p,1}_{i_j,i,i_k})$ except for a pair $P^{q-1}_{i_j,i,i_k}$ and $P^q_{i_j,i,i_k}$

for some q: the two subgraphs $P^{q-1}_{i_j,i,i_k}$ and $P^{q}_{i_j,i,i_k}$ are connected by a path of length four $\langle w^{q-1,3}_{i_j,i,i_k}, \beta^1_{i_j,i,i_k}, \beta^2_{i_j,i,i_k}, \beta^3_{i_j,i,i_k}, w^{q,1}_{i_j,i,i_k} \rangle$. This q can be arbitrary since we just want to insert the path of length four into the path gadget, and as an example, $q = 3$ in the path gadget P_{i_1,i,i_4} in Fig. 3.3. Finally, we prepare a special vertex α_i, and α_i is connected to all $\{\beta^1_{i_l,i,i_k}, \beta^2_{i_l,i,i_k}, \beta^3_{i_l,i,i_k}\}$'s. In the following, α_1, α_2, $\cdots$, α_n are called *α-vertices*. Similarly, β-vertices and γ-vertices mean the vertices labeled by β and γ, respectively. Since each path gadget has $4n^3 + 3$ vertices (two of which are shared with other path gadgets), the total number of vertices in H_i is

$$|V(H_i)| = \frac{deg(v_i)(deg(v_i) - 1)(4n^3 + 1)}{2} + deg(v_i) + 1,$$

i.e., there are $O(n^5)$ vertices in H_i.

(ii) Next we explain construction of the edge sets E_1 through E_m. Now suppose that e_k connects v_i with v_j for $i \neq j$. Also suppose that the sets of vertices adjacent to v_i and v_j are $N(v_i) = \{j, i_2, \cdots, i_{deg(v_i)}\}$ and $N(v_j) = \{i, j_2, \cdots, j_{deg(v_j)}\}$, respectively. Then, $(u_{i,j}, u_{j,i}) \in E_k$ where $u_{i,j} \in V(H_i)$ in the ith subgraph H_i and $u_{j,i} \in V(H_j)$ in the jth subgraph H_j. Furthermore, by the following rules, γ-vertices in the path gadgets are connected: See Figure 3.4. No vertex other than $u_{i,j}$ in the path gadget $P_{x,i,y}$ for $x = j$ or $y = j$ in H_i is connected to any vertex in H_j. Similarly, no vertex other than $u_{j,i}$ in the path gadget $P_{s,j,t}$ for $s = i$ or $t = i$ in H_j is connected to any vertex in H_i. For a path gadgets $P_{x,i,y}$ in H_i, where $j \notin \{x, y\}$ we prepare a set of edges as follows. Let $D = \min_{k \in \{i,j\}}\{deg(v_k)(deg(v_k) - 1)/2 - (deg(v_k) - 1)\}$.

- In $P_{x,i,y}$, there are n^3 γ-vertices, $\gamma^1_{x,i,y}$ through $\gamma^{n^3}_{x,i,y}$. Consider D γ-vertices among those n^3 γ-vertices, the $((j-1)n^2 + 1)$th vertex $\gamma^{(j-1)n^2+1}_{x,i,y}$ through the $((j-1)n^2 + D)$th vertex $\gamma^{(j-1)n^2+D}_{x,i,y}$.

- Next take a look at the jth subgraph H_j and the path gadgets $P_{s,j,t}$'s for $i \notin \{s, t\}$. Note that the number of such gadgets is $deg(v_j)(deg(v_j) - 1)/2 - (deg(v_j) - 1)$ and hence at least D. Then, consider the $((i-1)n^2 + 1)$th vertex $\gamma^{(i-1)n^2+1}_{s,j,t}$ in each $P_{s,j,t}$. Here, the term "+1" in the superscript of γ comes from the assumption that $j_1 = i$; if $j_k = i$, we consider the $((i-1)n^2 + k)$th γ-vertex.

- Then, we can choose any function f which assigns each element in $\{1, \ldots, D\}$ to a string s, j, t such that $i \notin \{s, t\}$ and it holds $f(b) \neq f(c)$ if $b \neq c$. Finally, we connect $\gamma_{x,i,y}^{(j-1)n^2+k}$ with $\gamma_{f(k)}^{(i-1)n^2+1}$ for $1 \leq k \leq D$. It is important that the path gadget $P_{x,i,y}$ is connected to $P_{s,j,t}$ via only one edge.

Each subgraph H_i has $O(n^5)$ vertices and thus the total number of vertices $|V(H)| = O(n^6)$. Clearly, this reduction can be done in polynomial time. In the next two subsections, we show that both conditions (C1) and (C2) are satisfied by the above reduction.

Proof of Condition (C1)

Without loss of generality, suppose that a longest induced cycle in G is $C^* = \langle v_1, v_2, \cdots, v_\ell, v_1 \rangle$ of length ℓ, and thus $OPT_2(G) = |C^*| = \ell \geq g(n)$. Then we select the following subset S of $4(n^3 + 1) \times \ell$ vertices and the induced subgraph $G[S]$:

$$\begin{aligned} S = {} & V(P_{\ell,1,2}) \cup \{\alpha_1\} \cup V(P_{1,2,3}) \cup \{\alpha_2\} \\ & \cup \cdots \cup V(P_{\ell-1,\ell,1}) \cup \{\alpha_\ell\}. \end{aligned}$$

For example, take a look at the graph G illustrated in Figure 3.2 again. One can see that the longest induced cycle in G is $\langle v_1, v_3, v_4, v_2, v_1 \rangle$. Then, we select the connected subgraph induced on the following set of vertices:

$$\begin{aligned} & V(P_{2,1,3}) \cup \{\alpha_1\} \cup V(P_{1,3,4}) \cup \{\alpha_3\} \\ & \cup V(P_{2,4,3}) \cup \{\alpha_4\} \cup V(P_{1,2,4}) \cup \{\alpha_2\} \end{aligned}$$

It is easy to see that the induced subgraph is 3-regular and connected. Hence, the reduction satisfies the condition (C1).

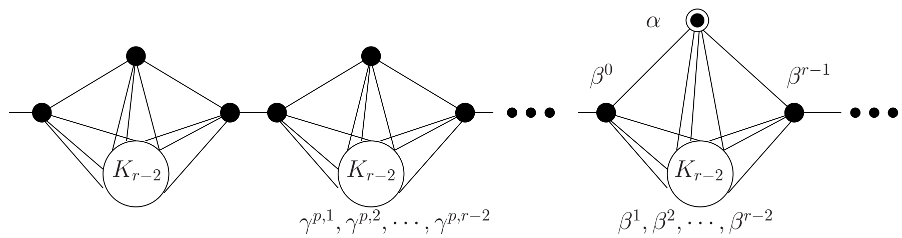


Fig. 3.5 Modified path gadget in the proof of Corollary 1

Proof of Condition (C2)

We show that the reduction satisfies the condition (C2) by showing its contraposition. Suppose that $OPT_3(H) \geq 4(n^3+1) \cdot \frac{g(n)}{n^{1-\varepsilon}}$ holds for a positive constant ε, and S^* is an optimal set of vertices such that the subgraph $H[S^*]$ induced on S^* is connected and 3-regular. In the following, one of the crucial observations is that we can select at most one path gadget from each subgraph H_i into the optimal set S^* of vertices, and if a portion of the path gadget is only selected, then the induced subgraph cannot be 3-regular.

(I) See Figure 3.3 again. Suppose for example that two path gadgets P_{i_1,i,i_4} and P_{i_2,i,i_3} are selected, and put their vertices into S^*. In order to make the degree of β-vertices three, we need to also select α_i. However, the degree of α_1 becomes six. This implies that we can select at most three β-vertices from each subgraph H_i.

(II) From the above observation (I), we consider the case that at most two of $\beta^1_{j,i,k}$, $\beta^2_{j,i,k}$, and $\beta^3_{j,i,k}$ are selected for some i, j, k. Let us assume that we select $\beta^1_{j,i,k}$ and $\beta^2_{j,i,k}$ ($\beta^1_{j,i,k}$ and $\beta^3_{j,i,k}$, resp.) are put into S^*, but $\beta^3_{j,i,k}$ ($\beta^2_{j,i,k}$, resp.) is not selected. Then, the degree of $\beta^2_{j,i,k}$ ($\beta^1_{j,i,k}$ and $\beta^3_{j,i,k}$, resp.) is at most 2 even if we select α_i, i.e., the induced subgraph cannot be 3-regular. By a similar reason, we cannot select only one of the β-vertices. Hence, if we select β-vertices, all of the three β-vertices in one path gadget must be selected.

As for w-vertices, a similar discussion can be done: For example, if we select $w^{p,1}_{j,i,k}$ and $w^{p,3}_{j,i,k}$ for some i, j, k, p, but $w^{p,2}_{j,i,k}$ ($\gamma^{p}_{j,i,k}$, resp.) is not selected, then the degree of $\gamma^{p}_{j,i,k}$ ($w^{p,2}_{j,i,k}$, resp.) is only 2. Thus, we need to select all the vertices of the part $P^{p}_{k,i,j}$ if we select some vertices from it.

Combining two observations above, one can see that the edges connecting $P^{p-1}_{k,i,j}$ and $P^{p}_{k,i,j}$, or w-vertices and β-vertices are necessary to make the degrees of the vertices three. As a result, we can conclude that if only a part of one path gadget is chosen, then the induced subgraph obtained cannot be 3-regular.

(III) From (I) and (II), we can assume that if some vertices of a path gadget are selected into S^*, it means that all vertices of the path gadget are selected. For example, suppose that P_{i_1,i,i_4} is selected. Since the degree of the endpoint u_{i,i_1} (u_{i,i_4}) of P_{i_1,i,i_4} is only 2, we have to put at least one vertex into S^* from another subgraph adjacent to H_i, say, a vertex $u_{j,i}$ in H_j. This implies that the induced subgraph $H[S^*]$ forms a cycle-like structure $\langle H_{i_1}, H_{i_2}, \cdots, H_{i_j}, H_{i_1}\rangle$ connecting $H_{i_1}, H_{i_2}, \cdots, H_{i_j}, H_{i_1}$ in order, where $\{i_1, i_2, \cdots, i_j\} \subseteq \{1, 2, \cdots, n\}$.

We mention that such an induced subgraph $H[S^*]$ is 3-regular if and only if the corresponding subgraph in the original graph G is an induced cycle. The if-part is clear by the discussion of the previous section. Let us look at the induced subgraph $H[V(P_{2,1,3}) \cup V(P_{1,3,4}) \cup V(P_{3,4,5}) \cup V(P_{2,5,4}) \cup V(P_{1,2,5})]$ in the right graph H shown in Figure 3.2. Then, the induced subgraph includes the edge (γ_1, γ_2) and thus the degrees of γ_1 and γ_4 are 4. The reason why the induced subgraph cannot be 3-regular comes from the fact that the cycle $\langle v_1, v_3, v_4, v_5, v_2, v_1\rangle$ includes the chord edge (v_1, v_4) in the original graph G. The edges between γ-vertices are placed because there is an edge between their corresponding vertices in G. As a result, the assumption that $H[S^*]$ is an optimal solution, i.e., 3-regular, implies that the corresponding induced subgraph in the original graph G forms a cycle $\langle v_{i_1}, v_{i_2}, \cdots, v_{i_j}, v_{i_1}\rangle$.

Since the number of vertices in each path gadget is $4(n^3+1)$, $OPT_2(G) \geq \frac{g(n)}{n^{1-\varepsilon}}$ holds by the assumption $OPT_3(H) \geq 4(n^3+1) \cdot \frac{g(n)}{n^{1-\varepsilon}}$. Therefore, the condition (C2) is also satisfied.

3.2.2 Reduction for $r \geq 4$

In this section, we give a brief sketch of the ideas to prove Corollary 1, i.e., the $O(n^{1/6-\varepsilon})$ inapproximability for r-MaxRICS for any fixed integer $r \geq 4$.

The proof is very similar to that of Theorem 1. The main difference between those proofs is the structure of each path gadget. See Figure 3.8, which shows the modified path gadget. (i) We replace each of γ-vertices in Figure 3.3 with the complete graph K_{r-2} of $r-2$ vertices, and then connect one γ-vertex in H_i and one γ-vertex in H_j for $i \neq j$ by a similar manner to the reduction for the case $r = 3$. (ii) As for β-vertices, we prepare K_{r-2} of $r-2$ vertices, say, $\beta^1, \cdots, \beta^{r-2}$, and two vertices, say, β^0 and β^{r-1}, such that each of the two vertices β^0 and β^{r-2} is adjacent to all the vertices in K_{r-2}. Then, all of the β-vertices are connected to the α-vertex similar to the reduction for $r = 3$. Since the reduction requires n^3 γ-vertices to connect all the pairs of H_i's, which is independent of the value of r, the path gadget consists of $\lceil \frac{n^3}{r-2} \rceil$ subgraphs, say, $P^1_{j,i,k}$ through $P^{\lceil n^3/(r-2) \rceil}_{j,i,k}$. As a result, the total number of vertices in the constructed graph remains $O(n^6)$. This completes the proof and thus we can obtain the $n^{1/6-\varepsilon}$ inapproximability of the general r-MaxRICS problem for $r \geq 4$.

3.3 Improved Hardness of Approximating r-MaxRICS

In this section we give the proof of Theorem 2. The hardness of approximating r-MaxRICS for $r \geq 3$ is shown via a gap-preserving reduction from MaxIS. First we show the $O(n^{1/2-\varepsilon})$-inapproximability of 3-MaxRICS and then the same $O(n^{1/2-\varepsilon})$-inapproximability of the general r-MaxRICS problem for any fixed integer $r \geq 4$. Consider an input graph $G = (V(G), E(G))$ of MaxIS with n vertices. Then, we construct a graph $H = (V(H), E(H))$ of 3-MaxRICS with $(4n + 8) \times n = 4n^2 + 8n$ vertices.

Let $OPT_1(G)$ (and $OPT_2(H)$, respectively) denote the number of vertices of an optimal solution for G of MaxIS (and H of 3-MaxRICS, respectively). Let $V(G) = \{v_1, v_2, \cdots, v_n\}$ of n vertices and $E(G) = \{e_1, e_2, \cdots, e_m\}$ of m edges. Let $g(n)$ be a parameter function of the instance G. Then we provide the gap-preserving reduction such that (C1) if $OPT_1(G) \geq g(n)$, then $OPT_2(H) \geq (4n + 8) \times g(n)$, and (C2) if $OPT_1(G) < \frac{g(n)}{n^{1-\varepsilon}}$ for a positive constant ε, then $OPT_2(H) < (4n + 8) \times \frac{g(n)}{n^{1-\varepsilon}}$. As we will explain later, the number of vertices in the reduced graph H is $O(n^2)$. Hence the approximation gap is $n^{1-\varepsilon} = O(|V(H)|^{1/2-\varepsilon})$ for any constant $\varepsilon > 0$.

Theorem 2 *For any fixed integer $r \geq 3$, r-MaxRICS on graphs of n vertices cannot be approximated in polynomial time within a factor of $n^{1/2-\varepsilon}$ for any constant $\varepsilon > 0$ if $\mathcal{P} \neq \mathcal{NP}$.*

3.3.1 Reduction for $r = 3$

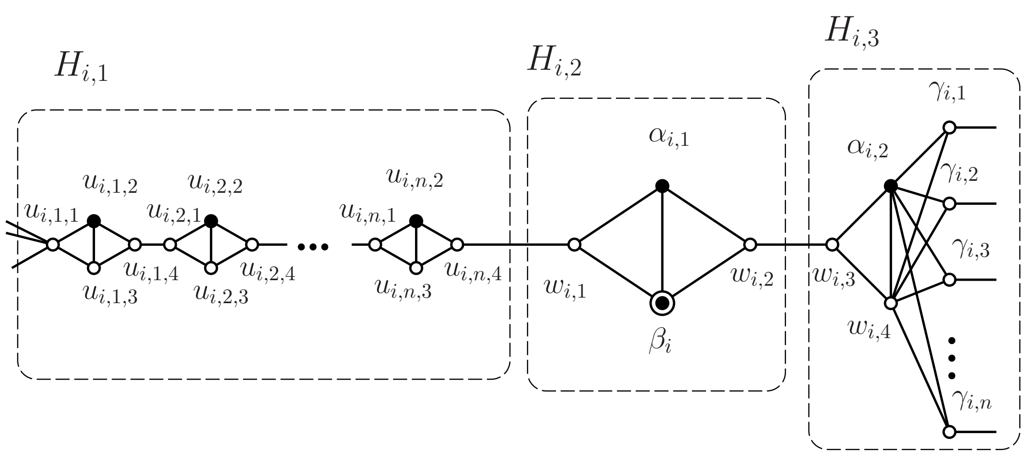


Fig. 3.6 Subgraph H_i

(1) The constructed graph H consists of n subgraphs, H_1 through H_n, where every subgraph $H_i = (V(H_i), E(H_i))$ is identical. Here we describe the construction of the ith subgraph H_i in detail for some i ($1 \leq i \leq n$). Figure 3.14 illustrates H_i, which is further divided into three subgraphs, $H_{i,1}$, $H_{i,2}$, and $H_{i,3}$.

(i) The leftmost subgraph $H_{i,1}$ forms a diamond-path, which is called a *path gadget*, and it includes the following $4 \times n = 4n$ vertices:

$$V(H_{i,1}) = \{u_{i,1,1}, u_{i,1,2}, u_{i,1,3}, u_{i,1,4}\} \cup \{u_{i,2,1}, u_{i,2,2}, u_{i,2,3}, u_{i,2,4}\} \cup \cdots \cup \{u_{i,n,1}, u_{i,n,2}, u_{i,n,3}, u_{i,n,4}\}$$

For $j = 1, 2, \cdots, n$, the subgraph induced on four vertices $\{u_{i,j,1}, u_{i,j,2}, u_{i,j,3}, u_{i,j,4}\}$ is a diamond graph, i.e., it has five edges, $(u_{i,j,1}, u_{i,j,2})$, $(u_{i,j,1}, u_{i,j,3})$, $(u_{i,j,2}, u_{i,j,3})$, $(u_{i,j,2}, u_{i,j,4})$ and $(u_{i,j,3}, u_{i,j,4})$. For $j = 1, 2, \cdots, n-1$, $u_{i,j,4}$ is connected to $u_{i,j+1,1}$.

(ii) The middle subgraph $H_{i,2}$ is called a *vertex gadget* and it is a diamond graph including 4 vertices and 5 edges, i.e.,

$$V(H_{i,2}) = \{w_{i,1}, \alpha_{i,1}, \beta_i, w_{i,2}\},$$

$$E(H_{i,2}) = \{(w_{i,1}, \alpha_{i,1}), (w_{i,1}, \beta_i), (\alpha_{i,1}, \beta_i), (\alpha_{i,1}, w_{i,2}), (\beta_i, w_{i,2})\}.$$

(iii) The leftmost subgraph $H_{i,3}$ is called a *connector gadget* and has the following $3 + n$ vertices:

$$V(H_{i,3}) = \{w_{i,3}, w_{i,4}, \alpha_{i,2}\} \cup \{\gamma_{i,1}, \gamma_{i,2}, \cdots, \gamma_{i,n}\}$$

There are three edges $(w_{i,3}, \alpha_{i,2})$, $(\alpha_{i,2}, w_{i,4})$ and $(w_{i,4}, w_{i,3})$ in $E(H_{i,3})$. Furthermore, for $j = 1, 2, \cdots, n$, the vertices $\alpha_{i,2}$ and $w_{i,4}$ are connected to $\gamma_{i,j}$. Here, the subgraph induced on four vertices $\{w_{i,3}, w_{i,4}, \alpha_{i,2}, \gamma_{i,j}\}$ for every j is again a diamond graph.

(iv) The path gadget $H_{i,1}$ and the vertex gadget $H_{i,2}$ are connected by an edge $(u_{i,n,4}, w_{i,1})$ and $H_{i,2}$ and the connector gadget $H_{i,3}$ are connected by an edge $(w_{i,2}, w_{i,3})$.

(2) Next we explain how we connect n subgraphs H_1 through H_n. (i) The leftmost vertex $u_{i,1,1}$ in the ith subgraph H_i is connected to the rightmost n vertices $\gamma_{1,i} \in V(H_1)$, $\gamma_{2,i} \in V(H_2)$, $\cdots$, $\gamma_{n,i} \in V(H_n)$ for each $i = 1, 2, \cdots, n$. (ii) If edge $(v_i, v_j) \in E(G)$, then we add an edge (β_i, β_j) for any pair i and j. Note that the induced graph $G[\{\beta_1, \beta_2, \cdots, \beta_n\}]$ is identical to the input graph G of MaxIS. This completes the reduction. It is clear that the reduction can be done in polynomial time.

Just to make the above construction clear, see Figure 3.7. For example, if an input instance G is illustrated in Figure 3.7-(a), then the reduced graph H is in Figure 3.7-(b), where some details are omitted due to the space. (i) The leftmost vertex $u_{1,1,1}$ in the top subgraph in H is connected to the rightmost vertices $\gamma_{1,1}$ in the top one, $\gamma_{2,1}$ in the second one, and so on. The vertex $u_{5,1,1}$ in the bottom is connected to five vertices $\gamma_{1,5}$ through $\gamma_{5,5}$. (ii) For example, since two vertices v_1 and v_2 are connected by an edge (v_1, v_2) in G, the vertex β_1 in the top vertex gadget $H_{1,2}$ is connected to the vertex β_2 in the second vertex gadget $H_{2,2}$. Also, according to an edge (v_1, v_5) in G, we add an edge (β_1, β_5) between $H_{1,2}$ and $H_{5,2}$.

Proof of Condition (C1)

Suppose that the graph G of MaxIS has the maximum independent set $IS^* = \{v_{1^*}, v_{2^*}, \cdots, v_{k^*}\}$ of size k, where $\{1^*, 2^*, \cdots, k^*\} \subseteq \{1, 2, \cdots, n\}$ and $1^* < 2^* < \cdots < k^*$. Also suppose that $OPT_1(G) = |IS^*| = k \geq g(n)$ holds. Then we select the following subset $S = S_{1^*} \cup S_{2^*} \cup \cdots \cup S_{k^*}$ of $(4n + 8) \times k \geq (4n + 8) \times g(n)$ vertices and the induced subgraph $H[S]$ from the constructed graph H:

$$
\begin{aligned}
S_{1^*} &= V(H_{1^*,1}) \cup \{\alpha_{1^*,1}, w_{1^*,1}, w_{1^*,2}, \beta_{1^*}\} \cup \{w_{1^*,3}, w_{1^*,4}, \alpha_{1^*,2}, \gamma_{1^*,2^*}\}, \\
S_{2^*} &= V(H_{2^*,1}) \cup \{\alpha_{2^*,1}, w_{2^*,1}, w_{2^*,2}, \beta_{2^*}\} \cup \{w_{2^*,3}, w_{2^*,4}, \alpha_{2^*,2}, \gamma_{2^*,3^*}\}, \\
&\vdots \\
S_{k^*} &= V(H_{k^*,1}) \cup \{\alpha_{k^*,1}, w_{k^*,1}, w_{k^*,2}, \beta_{k^*}\} \cup \{w_{k^*,3}, w_{k^*,4}, \alpha_{k^*,2}, \gamma_{k^*,1^*}\}.
\end{aligned}
$$

That is, for every $i \in \{1^*, 2^*, \cdots, k^*\}$, all the $4n$ vertices in the path gadget $H_{i,1}$, one diamond of 4 vertices in the vertex gadget $H_{i,2}$, and one diamond of 4 vertices in the connector gadget $H_{i,3}$ are selected.

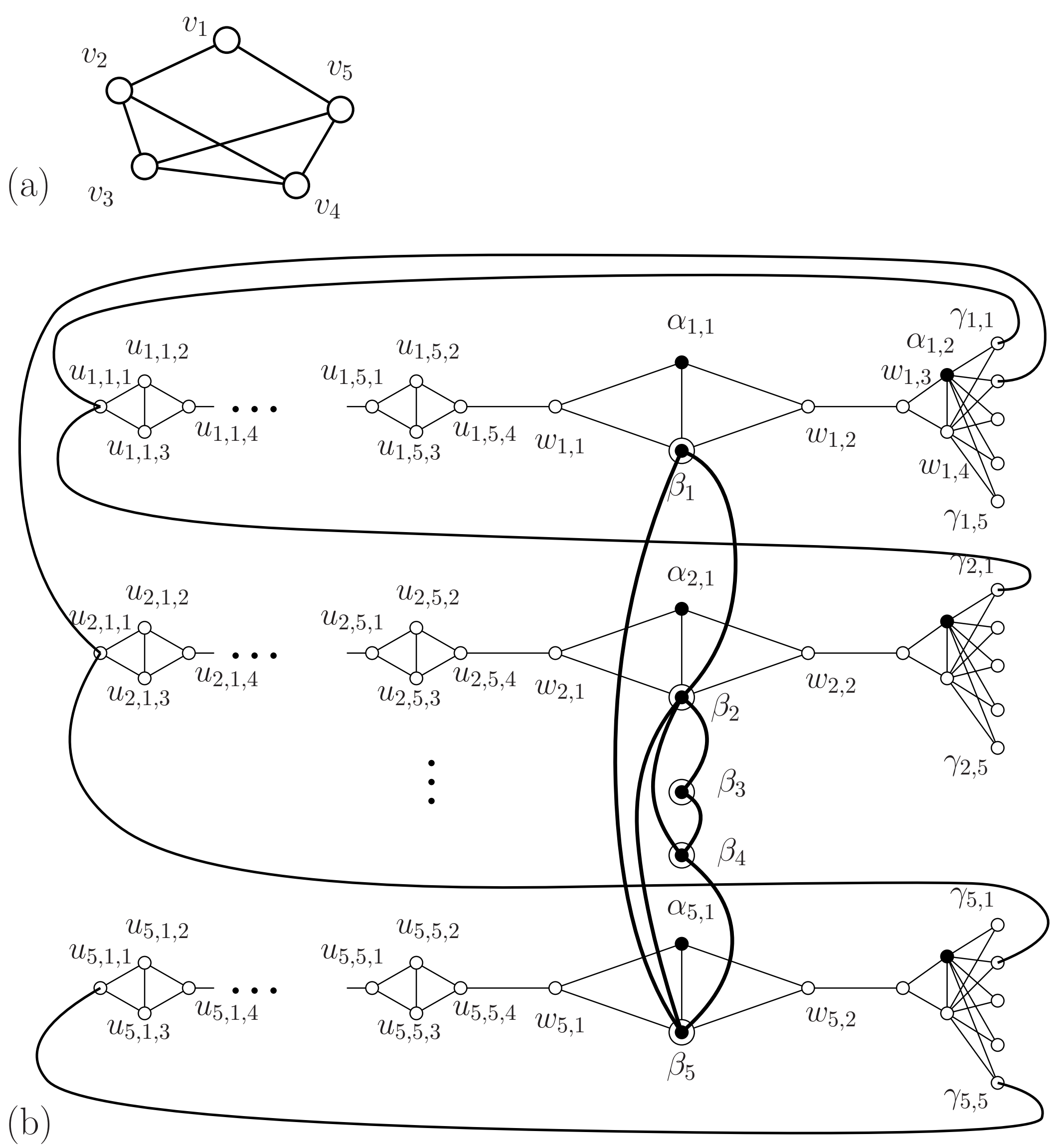


Fig. 3.7 (a) graph G of MaxIS, and (b) reduced graph H from G

For example, take a look at the graph G illustrated in Figure 3.7 again. One can see that the subset $\{v_1, v_3\}$ is one of the maximum independent sets in G. Then, we set $1^* = 1$ and $2^* = 3$ and thus select the connected subgraph induced on the following set S of $8n + 16$ vertices:

$$\begin{aligned} S = &V(H_{1,1}) \cup \{\alpha_{1,1}, w_{1,1}, w_{1,2}, \beta_1\} \cup \{\alpha_{1,2}, w_{1,3}, w_{1,4}, \gamma_{1,3}\} \cup \\ &V(H_{3,1}) \cup \{\alpha_{3,1}, w_{3,1}, w_{3,2}, \beta_3\} \cup \{\alpha_{3,2}, w_{3,3}, w_{3,4}, \gamma_{3,1}\} \end{aligned}$$

It is easy to see that the induced subgraph $H[S]$ is a diamond-cycle. It follows that $H[S]$ is 3-regular and connected. Hence, the reduction satisfies the condition (C1).

Proof of Condition (C2)

We show that the above reduction satisfies the condition (C2) by showing its contraposition, i.e., if $OPT_2(H) \geq (4n + 8) \times \frac{g(n)}{n^{1-\varepsilon}}$, then $OPT_1(G) \geq \frac{g(n)}{n^{1-\varepsilon}}$ for a positive constant ε. Now suppose that $OPT_2(H) \geq (4n + 8) \cdot \frac{g(n)}{n^{1-\varepsilon}}$ holds for a positive constant ε, and S^* is an optimal set of vertices such that the subgraph $H[S^*]$ induced on S^* is connected and 3-regular. Let $|S^*| = (4n + 8) \times k = (4n + 8) \cdot \lceil \frac{g(n)}{n^{1-\varepsilon}} \rceil$.

(1) Consider the ith subgraph H_i. For example, if we select $4n$ vertices in the path gadget $H_{i,1}$, four vertices $\alpha_{i,1}$, $w_{1,1}$, $w_{1,2}$, β_i in the vertex gadget $H_{i,2}$, and four vertices $\alpha_{i,2}$, $w_{i,3}$, $w_{i,4}$, $\gamma_{i,i}$ in the connector gadget $H_{i,3}$, then the subgraph induced on those $4n + 8$ vertices is 3-regular and connected. Therefore, $|S^*| \geq 4n + 8$ always holds.

(2) Take a look at the ith path gadget $H_{i,1}$. The degree of every vertex, say, u, in $H_{i,1}$ except for the head vertex $u_{i,1,1}$ and the tail vertex $u_{i,n,4}$ is 3. Thus, if S^* includes the vertex u such that $deg(u) = 3$ in $H_{i,1}$, then S^* includes the whole set $V(H_i, 1)$ of $4n$ vertices, i.e, $V(H_{i,1}) \subseteq S^*$ holds. Furthermore, the neighbor vertex $w_{i,1}$ of the tail $u_{i,n,4}$ also must be in S^* since $deg(u_{i,n,4}) = 3$.

(3) Next consider the vertex gadget $H_{i,2}$. Similarly to the above observation (2), if at least one vertex in $\{\alpha_{i,1}, w_{i,1}, w_{i,2}\}$ is in the solution S^*, then four vertices $\alpha_{i,1}, \beta_i$, $w_{i,1}$ and $w_{i,2}$ must be selected into S^* since $deg(\alpha_{i,1}) = deg(w_{i,1}) = deg(w_{i,2}) = 3$. In the case that $V(H_{i,2}) \subseteq S^*$, the vertex $\beta_i \in V(H_{i,2})$ cannot be connected to any other vertices in the different subgraph, say, $H_{j,2}$ such that $i \neq j$. Even if only β_i is selected into S^*, the total number of vertices selected from $\{\beta_1, \beta_2, \cdots, \beta_n\}$ is at most n.

(4) It can be shown that exactly four vertices which form a diamond graph can be selected from the connector gadget $H_{i,3}$ into S^* in order to make the degree of vertices 3.

From the above observation (1) through (4), we can select at most $4n + 4 + 4 = 4n + 8$ vertices from each subgraph H_i into the solution S^*, where $4n$, 4 and 4 come from the path gadget $H_{i,1}$, the vertex one $H_{i,2}$ and the connector one $H_{i,2}$, respectively, and the subgraph induced on those $4n + 8$ vertices is a diamond-path. By the assumption $|S^*| = (4n + 8) \times k$, we must select at least k diamond-paths of $4n + 8$ vertices as an induced subgraph of $H[S^*]$.

Recall that the diamond-path of $4n + 8$ vertices in each subgraph H_i surely includes β_i. Here, it is important to note that the vertex β_i has three neighbors $\alpha_{i,1}$, $w_{i,1}$, and $w_{i,2}$ as mentioned before. That is, in the induced subgraph $H[S^*]$, we can find an independent set of k vertices labeled by β's, which corresponds to an independent set of k vertices in the input graph G of MaxIS. As are result, $OPT_1(G) \geq k \geq \frac{g(n)}{n^{1-\varepsilon}}$ holds for a positive constant ε by the assumption $OPT_2(H) = (4n + 8) \times k \geq (4n + 8) \times \frac{g(n)}{n^{1-\varepsilon}}$. Therefore, the condition (C2) is also satisfied.

3.3.2 Reduction for $r \geq 4$

In this section, we give a brief sketch of the ideas to prove the $O(n^{1/2-\varepsilon})$ inapproximability for r-MaxRICS for any fixed integer $r \geq 4$. The proof is very similar to that of $r = 3$. The main difference between those proofs is the structure of each subgraph H_i. See Figure 3.8, which shows the modified subgraph H_i. (i) We replace the previous vertex $u_{i,1,2}$ in Figure 3.14 with a complete graph K_{r-2} of $r - 2$ vertices, labeled by $U_{i,1,2}$. The three vertices $u_{i,1,1}$, $u_{i,1,3}$ and $u_{i,1,4}$ are connected to all the $r - 2$ vertices in $U_{i,1,2}$, respectively. Also $u_{i,2,2}$ is replaced with a complete graph of $r - 2$ vertices, and so on. But, we now prepare only $\lceil \frac{4n}{r+1} \rceil$ "modified" diamonds of $(r - 2) + 3 = r + 1$ vertices. Namely, $4n \leq |V(H_{i,1})| \leq 5n$ holds. (ii) In the vertex gadget $H_{i,2}$, the previous $\alpha_{i,1}$ is replaced with a complete graph $A_{i,1}$ of $r - 2$ vertices. Thus, the number of vertices in $V(H_{i,2})$ is $r + 1 \leq n$. The three vertices $w_{i,1}$, $w_{i,2}$ and β_i are connected to all the $r - 2$ vertices, respectively. (iii) In the connector gadget $H_{i,3}$, $\alpha_{i,2}$ is replaced with a complete graph $A_{i,2}$ of $r-2$ vertices. Thus, $|V(H_{i,3})| \le 2n$. As a result, the number of vertices in the modified subgraph H_i remains $O(n)$, which means that $|V(H)| = O(n^2)$. This completes the proof and thus we can obtain the $n^{1/2-\varepsilon}$ inapproximability of the general r-MaxRICS problem for $r \ge 4$.

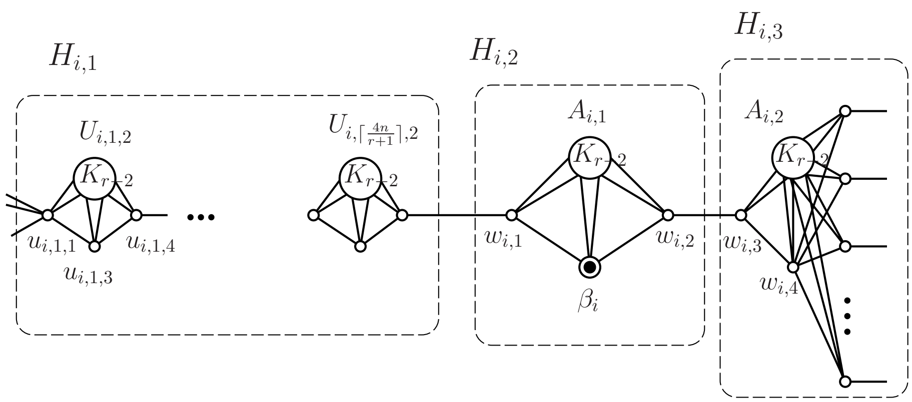


Fig. 3.8 Modified subgraph H_i in the case of $r \ge 4$

3.4 Further Improved Hardness Inapproximability of r-MaxRIS

In sections 3.4, 3.5 and 3.6, we thus consider the problems whose input graphs are restricted to some special classes of graphs. In the rest of this section, we give the complexity results. We first show that the problems are still NP-hard to approximate even if r is a fixed constant and the input graph is either bipartite or planar. Indeed, we consider the decision problem, called r-OneRIS, which determines whether a given graph G contains at least one r-regular induced subgraph or not. Note that r-OneRIS simply asks the existence of an r-regular induced subgraph in G, and hence this is a decision version of both r-MaxRIS and r-MaxRICS in the sense that the problem determines whether $\mathrm{OPT}_{\mathsf{RIS}}(G) > 0$ and $\mathrm{OPT}_{\mathsf{RICS}}(G) > 0$ hold or not. Clearly, r-OneRIS for $r = 0, 1, 2$ can be solved in linear time for any graph, because it simply finds one vertex, one edge and one induced cycle, respectively.

3.4.1 Bipartite graphs

In this subsection, we give the complexity result for bipartite graphs. Since r-OneRIS for $r = 0, 1, 2$ can be solved in linear time, the following theorem gives the dichotomy result for bipartite graphs.

Theorem 3 *For every fixed integer $r \geq 3$, r-OneRIS is NP-complete for bipartite graphs of maximum degree $r + 1$.*

Proof of Theorem 3.

It is obvious that r-OneRIS belongs to NP. Therefore, we show that r-OneRIS is NP-hard for bipartite graphs of maximum degree $r + 1$ by giving a polynomial-time reduction from the following decision problem (in which the induced property is *not* required): the problem r-OneRS is to determine whether a given graph H contains at least one r-regular subgraph or not. It is known that r-OneRS is NP-complete even if $r = 3$ and the input is a bipartite graph of maximum degree four [26].

[Main ideas of our reduction]

We now explain our ideas of the reduction. Let H be a bipartite graph of maximum degree four as an instance of 3-OneRS, and let G_H be the bipartite graph of maximum degree $r + 1$ which corresponds to H as the instance of r-OneRIS. The construction of G_H will be given later, but G_H is constructed so that H contains a 3-regular subgraph if and only if G_H contains an r-regular induced subgraph. In r-OneRS, we can decide whether an edge of H is contained in a solution or not. On the other hand, since r-OneRIS requires the induced property, we are *not* given such a choice for edges in r-OneRIS; we can select only vertices of G_H to construct an r-regular induced subgraph. Therefore, the key point of our reduction is how to simulate a selection of an edge of H by choosing vertices of G_H.

We first show that 3-OneRIS is NP-hard for bipartite graphs of maximum degree four, and then modify the reduction for $r = 3$ to general $r \geq 4$.

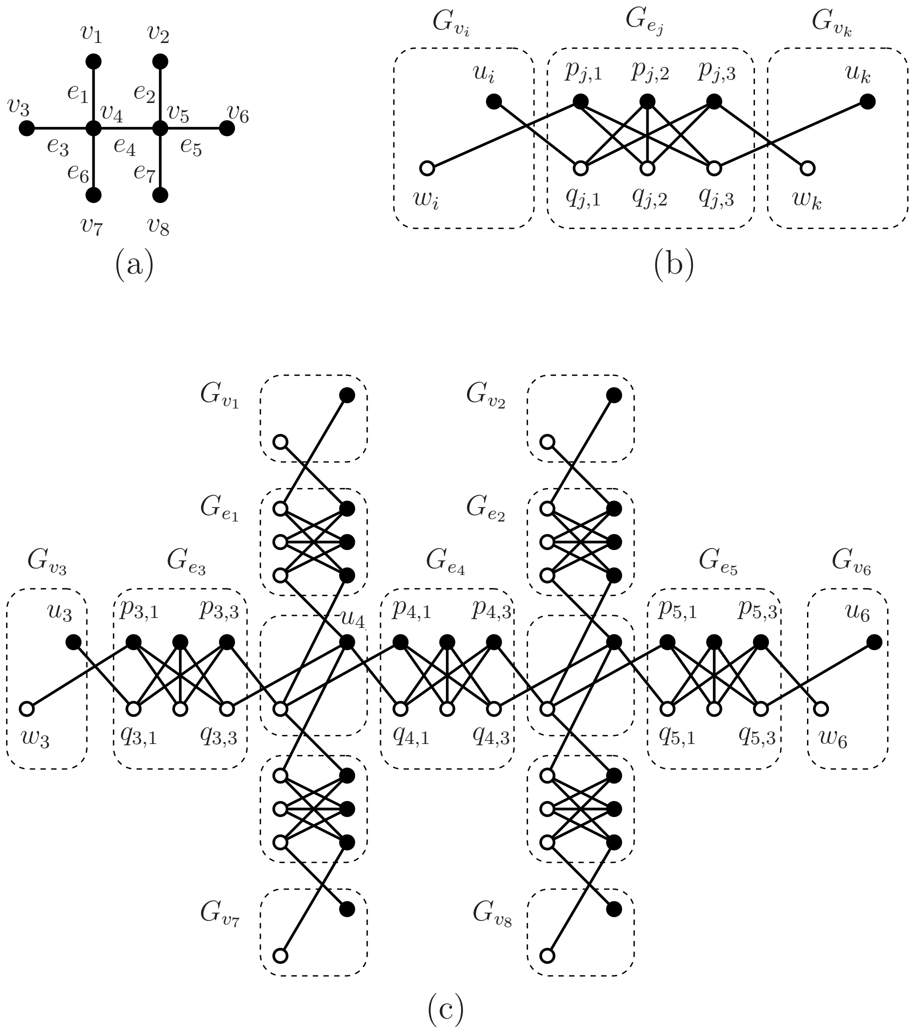


Fig. 3.9 (a) Input graph H of 3-OneRS, (b) three gadgets G_{v_i}, G_{e_j} and G_{v_k} corresponding to an edge $e_j = (v_i, v_k)$ in $E(H)$, and (c) the corresponding graph G_H of 3-OneRIS.

[Reduction for $r = 3$]

Let $V(H) = \{v_1, v_2, \ldots, v_n\}$ of n vertices, and $E(H) = \{e_1, e_2, \ldots, e_m\}$ of m edges. The corresponding graph G_H consists of

(i) n subgraphs $G_{v_1}, G_{v_2}, \ldots, G_{v_n}$, called *vertex-gadgets*, which are associated with n vertices $v_1, v_2, \ldots, v_n$ in $V(H)$, respectively;

(ii) m subgraphs $G_{e_1}, G_{e_2}, \ldots, G_{e_m}$, called *edge-gadgets*, which are associated with m edges

$e_1, e_2, \ldots, e_m$ in $E(H)$, respectively; and

(iii) the set of edges which connect vertex-gadgets and edge-gadgets.

Below we construct each gadget and the corresponding graph G_H. (See Fig. 3.9.)

(i) For each i, $1 \leq i \leq n$, the i-th vertex-gadget G_{v_i} consists only of two isolated vertices u_i and w_i, and hence $E(G_{v_i}) = \emptyset$.

(ii) For each j, $1 \leq j \leq m$, the j-th edge-gadget G_{e_j} can be obtained from a complete bipartite graph $K_{3,3}$ by deleting two edges, as follows: suppose that, in $K_{3,3}$, one side consists of three vertices $p_{j,1}$, $p_{j,2}$, $p_{j,3}$ and the other side consists of three vertices $q_{j,1}$, $q_{j,2}$, $q_{j,3}$; then, delete the two edges $(p_{j,1}, q_{j,1})$ and $(p_{j,3}, q_{j,3})$.

(iii) For each edge $e_j = (v_i, v_k)$ in $E(H)$ such that $i < k$, we connect the gadgets G_{v_i}, G_{e_j} and G_{v_k} by four edges, as follows: add two edges $(u_i, q_{j,1})$ and $(w_i, p_{j,1})$ between G_{v_i} and G_{e_j}, and also add two edges $(q_{j,3}, u_k)$ and $(p_{j,3}, w_k)$ between G_{e_j} and G_{v_k}.

This completes the construction of the corresponding graph G_H. Clearly, this reduction can be done in polynomial time. Furthermore, G_H is bipartite. (See Fig. 3.9(c) as an example; the set of white vertices and the set of black vertices form a bipartition of $V(G_H)$.)

By the construction, we have the following lemma.

Lemma 1 *The graph G_H satisfies the following (a) and (b).*

(a) *Consider an edge-gadget G_{e_j} corresponding to an edge $e_j = (v_i, v_k)$ in $E(H)$ such that $i < k$. If a 3-regular induced subgraph in G_H contains a vertex in G_{e_j}, then all vertices in G_{v_i}, G_{e_j} and G_{v_k} are contained in the subgraph.*

(b) *For a vertex-gadget G_{v_i}, $1 \leq i \leq n$, the vertex $u_i \in V(G_{v_i})$ is contained in a 3-regular induced subgraph in G_H if and only if the vertex $w_i \in V(G_{v_i})$ is contained in the subgraph.*

proof 1 *(a) Note that every vertex v in the j-th edge-gadget G_{e_j} is of degree exactly three in G_H, that is, $d(G_H, v) = 3$. Therefore, if a vertex $v \in V(G_{e_j})$ is contained in a 3-regular (induced) subgraph, then all vertices in $N(G_H, v)$ must be also contained in the subgraph.*

Then, since $u_i \in N(G_H, q_{j,1})$, $w_i \in N(G_H, p_{j,1})$, $u_k \in N(G_H, q_{j,3})$ and $w_k \in N(G_H, p_{j,3})$, all vertices in G_{v_i}, G_{e_j} and G_{v_k} are contained in the subgraph.

(b) Suppose that $u_i \in V(G_{v_i})$ is contained in a 3-regular induced subgraph G'_H of G_H. (The proof for the other direction is the same.) Since $d(G'_H, u_i) = 3$, exactly three vertices incident to u_i are contained in G'_H. Recall that $d(G_{v_i}, u_i) = 0$, and hence they must be vertices in edge-gadgets incident to G_{v_i}. Then, Lemma 1(a) implies that w_i is also contained in G'_H. □

Lemma 1(a) implies that a selection of an edge e_j of H can be simulated by choosing vertices of the corresponding edge-gadget G_{e_j} in G_H. Note that the vertices in G_{e_j} are *not* necessarily selected even if the vertices in $V(G_{v_i}) \cup V(G_{v_k})$ are selected; since each vertex in $V(G_{v_i}) \cup V(G_{v_k})$ is of degree at most four, it can be incident to three edge-gadgets other than G_{e_j}.

We now show that the graph H of 3-OneRS contains a 3-regular subgraph if and only if the corresponding graph G_H of 3-OneRIS contains a 3-regular induced subgraph.

Suppose that H contains a 3-regular subgraph H'. Then, we simply choose all vertices in the gadgets in G_H that correspond to the vertices in $V(H')$ and the edges in $E(H')$. Notice that $d(G_{v_i}, v) = 0$ for each vertex v in a vertex-gadget G_{v_i}, $1 \leq i \leq n$, and $d(G_H, v) = 3$ for each vertex v in an edge-gadget G_{e_j}, $1 \leq j \leq m$. Then, the subgraph of G_H induced by the chosen vertices is clearly 3-regular.

Conversely, suppose that G_H contains a 3-regular induced subgraph G'_H. Lemma 1 implies that G'_H contains either all vertices or none of the vertices of each gadget in G_H. Therefore, one can obtain a subgraph H' of H which corresponds to G'_H. Recall that $d(G_{v_i}, u_i) = d(G_{v_i}, w_i) = 0$ for the two vertices u_i and w_i in each vertex-gadget G_{v_i}, $1 \leq i \leq n$. Since G'_H is 3-regular, $d(G'_H, u_i) = d(G'_H, w_i) = 3$ if the two vertices u_i and w_i are contained in G'_H. Then, exactly three edge-gadgets incident to G_{v_i} must be contained in G'_H. This means that exactly three edges are incident to v_i in the subgraph H'. Therefore, H' is also 3-regular.

This completes the proof for $r = 3$.

[Reduction for $r \geq 4$]

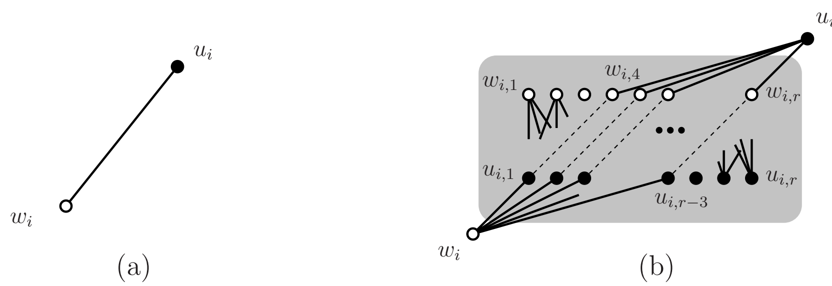


Fig. 3.10 Vertex-gadgets $G^r_{v_i}$ for (a) $r = 4$ and (b) $r \geq 5$. The internal graph of $G^r_{v_i}$, $r \geq 5$, is shaded.

In the following, we show that the reduction for $r = 3$ can be modified for $r \geq 4$. Let G^3_H be the graph G_H constructed above for $r = 3$; we denote by G^r_H the corresponding graph for r-OneRIS, $r \geq 4$.

The reduction for $r \geq 4$ is also given from 3-OneRS. Let H be a graph as an instance of 3-OneRS such that $V(H) = \{v_1, v_2, \ldots, v_n\}$ and $E(H) = \{e_1, e_2, \ldots, e_m\}$. The main differences between the reductions for $r = 3$ and $r \geq 4$ are the structures of vertex-gadgets $G^r_{v_i}$ and edge-gadgets $G^r_{e_j}$.

(i) For each i, $1 \leq i \leq n$,

- if $r = 4$, the vertex-gadget $G^4_{v_i}$ consists of a single edge (u_i, w_i) joining two vertices u_i and w_i, as illustrated in Fig. 3.10(a); and
- if $r \geq 5$, the vertex-gadget $G^r_{v_i}$ consists of two vertices u_i and w_i together with the *internal graph* between them, as illustrated in Fig. 3.10(b): the internal graph can be obtained from a complete bipartite graph $K_{r,r}$ by deleting $(r-3)$ edges $(w_{i,4}, u_{i,1})$, $(w_{i,5}, u_{i,2}), \ldots, (w_{i,r}, u_{i,r-3})$ forming a matching (they are illustrated as dotted lines in Fig. 3.10(b)); and then u_i is connected to every vertex in $\{w_{i,4}, w_{i,5}, \ldots, w_{i,r}\}$ and w_i is connected to every vertex in $\{u_{i,1}, u_{i,2}, \ldots, u_{i,r-3}\}$. Notice that all the vertices in the internal graph are of degree exactly r in $G^r_{v_i}$.

(ii) For each j, $1 \leq j \leq m$, the edge-gadget $G^r_{e_j}$ is a simple extension of $G^3_{e_j}$: it can be obtained by deleting two edges $(p_{j,1}, q_{j,1})$ and $(p_{j,r}, q_{j,r})$ from a complete bipartite graph $K_{r,r}$ of bipartition $\{p_{j,1},\ p_{j,2},\ldots,p_{j,r}\}$ and $\{q_{j,1},\ q_{j,2},\ldots,q_{j,r}\}$.

(iii) The connections between the vertex-gadgets and the edge-gadgets in G^r_H are the same as in G^3_H.

By the same arguments as in $r = 3$, the counterpart of Lemma 1 holds and hence we can simulate a selection of an edge e_j of H by choosing vertices of the j-th edge-gadget $G^r_{e_j}$. We thus focus on the vertex-gadgets $G^r_{v_i}$ such that $d(H, v_i) = r + 1$ below. (The arguments for $d(H, v_i) \leq r$ are similar.)

For $r = 4$, the vertices u_i and w_i in the vertex-gadget $G^4_{v_i}$ are of degree exactly 5, and hence only one edge is missing around u_i (or around w_i) in any 4-regular induced subgraph which contains u_i (resp., w_i). The counterpart of Lemma 1(b) implies that both u_i and w_i are always selected at the same time, and hence the missing edge cannot be the edge (u_i, w_i) due to the induced property. Therefore, it must be one of the four edges connecting to edge-gadgets; this ensures that the corresponding subgraph in H is 3-regular.

The arguments for $r \geq 5$ are almost the same. We now show that the counterpart of Lemma 1(b) holds for $r \geq 5$, that is, if one vertex in a vertex-gadget $G^r_{v_i}$ is contained in an r-regular induced subgraph of G^r_H, then all vertices in $G^r_{v_i}$ are contained in the subgraph. Firstly, if one vertex is selected from the internal graph of $G^r_{v_i}$, then all vertices in $G^r_{v_i}$ (including u_i and w_i) must be also selected; remember that all vertices in the internal graph are of degree exactly r. Secondly, consider the case where either u_i or w_i is selected. Recall that each of u_i and w_i is incident with exactly $(r + 1)$ edges, $(r - 3)$ of which are connecting to the internal graph of $G^r_{v_i}$. Since $r - 3 \geq 2$ and we have only one missing edge around u_i (or w_i), any r-regular induced subgraph in G^r_H contains at least one edge connecting to the internal graph of $G^r_{v_i}$. Then, the subgraph contains one vertex from the internal graph, and hence it must contain all vertices in $G^r_{v_i}$. In this way, the counterpart of Lemma 1(b) holds for $r \geq 5$. Therefore, the missing edge around u_i (or w_i) must be one of the four edges connecting to edge-gadgets; this ensures that the corresponding subgraph in H is 3-regular.

This completes the proof for $r \geq 4$. □

Theorem 3 implies the following corollary.

Corellary 2 *Let $\rho(n) \geq 1$ be any polynomial-time computable function. For every fixed integer $r \geq 3$ and bipartite graphs of maximum degree $r + 1$, r-*MaxRIS *and r-*MaxRICS *admit no polynomial-time approximation algorithm within a factor of $\rho(n)$ unless* P = NP.

proof 2 *We only give a proof for r-*MaxRIS*. (The proof for r-*MaxRICS *is the same.) Suppose for a contradiction that r-*MaxRIS *admits a polynomial-time $\rho(n)$-approximation algorithm for some polynomial-time computable function $\rho(n) > 0$. Then, the algorithm can compute a solution in polynomial time such that the objective value $\mathrm{APX}_{\mathsf{RIS}}(G)$ satisfies*

$$\mathrm{APX}_{\mathsf{RIS}}(G) \leq \mathrm{OPT}_{\mathsf{RIS}}(G) \leq \rho(n) \cdot \mathrm{APX}_{\mathsf{RIS}}(G).$$

Therefore, one can distinguish either $\mathrm{OPT}_{\mathsf{RIS}}(G) > 0$ or $\mathrm{OPT}_{\mathsf{RIS}}(G) = 0$ in polynomial time using the algorithm. This is a contradiction unless P = NP*, because Theorem 3 implies that it is NP-complete to determine whether $\mathrm{OPT}_{\mathsf{RIS}}(G) > 0$ or not if $r \geq 3$.* □

3.4.2 Planar graphs

In this subsection, we give the complexity result for planar graphs. Notice that Euler's formula implies that any 6-regular graph is not planar, and hence the answer to r-OneRIS is always "No" for planar graphs if $r \geq 6$. Therefore, the following theorem gives the dichotomy result for planar graphs.

Theorem 4 *For every fixed integer r, $3 \leq r \leq 5$, r-*OneRIS *is NP-complete for planar graphs.*

proof 3 *Since r-*OneRIS *belongs to NP, we show that r-*OneRIS *is NP-hard for planar graphs by giving a polynomial-time reduction from r-*OneRS*. For every fixed integer r, $3 \leq r \leq 5$, it is known that r-*OneRS *is NP-complete for planar graphs [24, 25]. It is important to notice that the reduction is made for the same value r.*

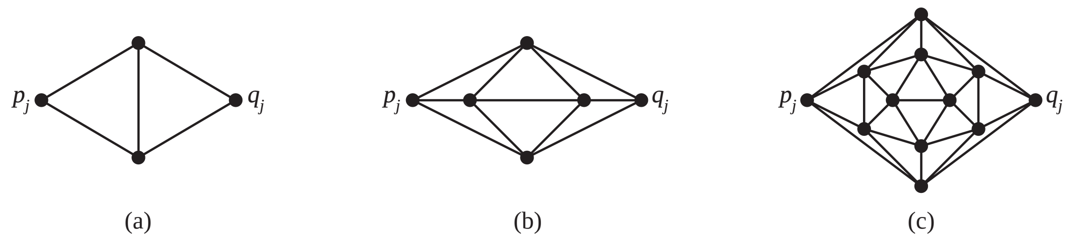


Fig. 3.11 Edge-gadgets (a) $G^3_{e_j}$ for $r = 3$, (b) $G^4_{e_j}$ for $r = 4$, and (c) $G^5_{e_j}$ for $r = 5$.

*Let r be a fixed integer such that $3 \le r \le 5$, and let H be a planar graph as an instance of r-*OneRS*. Then, the planar graph G^r_H corresponding to H is constructed as follows: for each edge $e_j = (v_i, v_k)$ in $E(H)$,*

- *we replace the edge e_j with the j-th* edge-gadget *$G^r_{e_j}$ which is given in Fig. 3.11; and*
- *connect $G^r_{e_j}$ to the two vertices v_i and v_k by two edges (v_i, p_j) and (q_j, v_k).*

Since H is planar, G^r_H is also planar. This construction can be clearly done in polynomial time. This completes the construction of the corresponding graph G^r_H.

*Similarly as in the proof of Theorem 3, the key point of our reduction is to simulate a selection of an edge $e_j = (v_i, v_k)$ of H by choosing vertices of the j-th edge-gadget $G^r_{e_j}$. It is important to notice that each vertex in $G^r_{e_j}$ is of degree exactly r in G^r_H. Therefore, if we select one vertex in $G^r_{e_j}$, then all vertices in $V(G^r_{e_j}) \cup \{v_i, v_k\}$ must be also selected. In contrast, the vertices in $G^r_{e_j}$ are not necessarily selected even if a vertex in $\{v_i, v_k\}$ is selected; it may be incident to r edge-gadgets other than $G^r_{e_j}$. Then, similar arguments as in the proof of Theorem 3 prove that the graph H for r-*OneRS *contains an r-regular subgraph if and only if the corresponding graph G^r_H for r-*OneRIS *contains an r-regular induced subgraph.*

This completes the proof of Theorem 4. □

The same arguments as in Corollary 2 establish the following corollary.

Corellary 3 *Let $\rho(n) \ge 1$ be any polynomial-time computable function. For every fixed integer r, $3 \le r \le 5$, r-*MaxRIS *and r-*MaxRICS *for planar graphs admit no polynomial-time approximation algorithm within a factor of $\rho(n)$ unless* P = NP.

3.5 Graphs with Bounded Treewidth

In this section, we consider the problems restricted to graphs with bounded treewidth. We first introduce the notion of treewidth in Section 3.5.1. Then, Section 3.5.2 gives a linear-time algorithm for r-MaxRIS. Section 3.5.3 shows that the algorithm for r-MaxRIS can be modified for r-MaxRICS.

3.5.1 Definitions

Let G be a graph with n vertices. A *tree-decomposition* of G is a pair $\langle\{X_i \mid i \in V_T\}, T\rangle$, where $T = (V_T, E_T)$ is a rooted tree, such that the following four conditions (1)–(4) hold [6]:

(1) each X_i is a subset of $V(G)$, and is called a *bag*;

(2) $\bigcup_{i \in V_T} X_i = V(G)$;

(3) for each edge $(u, v) \in E(G)$, there is at least one node $i \in V_T$ such that $u, v \in X_i$; and

(4) for each vertex $v \in V(G)$, the set $\{i \in V_T \mid v \in X_i\}$ induces a connected component in T.

We will refer to a *node* in V_T in order to distinguish it from a vertex in $V(G)$. The *width* of a tree-decomposition $\langle\{X_i \mid i \in V_T\}, T\rangle$ is defined as $\max\{|X_i| - 1 : i \in V_T\}$, and the *treewidth* of G is the minimum k such that G has a tree-decomposition of width k.

In particular, a tree-decomposition $\langle\{X_i \mid i \in V_T\}, T\rangle$ of G is called a *nice tree-decomposition* if the following four conditions (5)–(8) hold [4]:

(5) $|V_T| = O(n)$;

(6) every node in V_T has at most two children in T;

(7) if a node $i \in V_T$ has two children l and r, then $X_i = X_l = X_r$; and

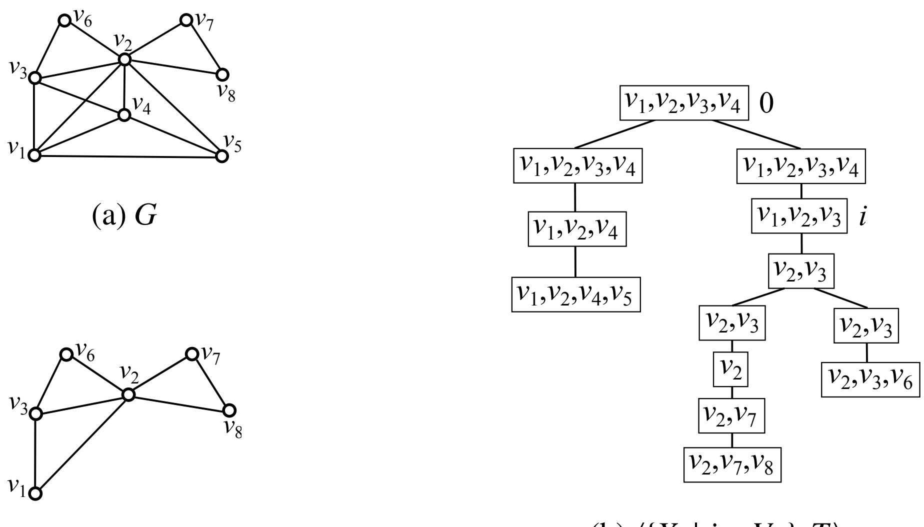
(8) if a node $i \in V_T$ has only one child j, then one of the following two conditions (a) and (b) holds:

    (a) $|X_i| = |X_j| - 1$ and $X_i \subset X_j$ (such a node i is called a *forget* node); and

    (b) $|X_i| = |X_j| + 1$ and $X_i \supset X_j$ (such a node i is called an *introduce* node.)

Figure 3.12(b) illustrates a nice tree-decomposition $\langle\{X_i \mid i \in V_T\}, T\rangle$ of the graph G in

Fig. 3.12(a) whose treewidth is three. It is known that any graph of treewidth k has a nice tree-decomposition of width k [4]. Since a nice tree-decomposition $\langle\{X_i \mid i \in V_T\}, T\rangle$ of a graph G with bounded treewidth can be found in linear time [4], we may assume without loss of generality that G and its nice tree-decomposition are both given.



(a) G

(b) $\langle\{X_i \mid i \in V_T\}, T\rangle$

(c) G_i

Fig. 3.12 (a) Graph G, (b) a nice tree-decomposition $\langle\{X_i \mid i \in V_T\}, T\rangle$ of G, and (c) the subgraph G_i of G for the node $i \in V_T$.

Each node $i \in V_T$ corresponds to a subgraph G_i of G which is induced by the vertices that are contained in the bag X_i and all bags of descendants of i in T. Therefore, if a node $i \in V_T$ has two children l and r in T, then G_i is the union of G_l and G_r which are the subgraphs corresponding to nodes l and r, respectively. Clearly, $G = G_0$ for the root 0 of T. For example, Fig. 3.12(c) illustrates the subgraph G_i of the graph G in Fig. 3.12(a) which corresponds to the node $i \in V_T$ in Fig. 3.12(b). By definitions (3) and (4) of a tree-decomposition, we have the following proposition.

Proposition 2 *For each node $i \in V_T$, there is no edge joining a vertex in $G_i \setminus X_i$ and one in $G \setminus G_i$.*

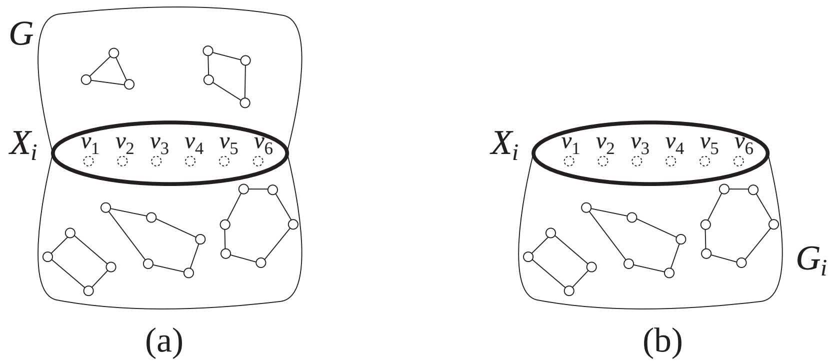


Fig. 3.13 (a) A 2-regular induced subgraph F of a graph G such that $V(F) \cap X_i = \emptyset$, and (b) the (K, ϕ)-subgraph F_i of G_i, where $X_i = \{v_1, v_2, \ldots, v_6\}$ and $K = \emptyset$.

3.5.2 Algorithm for r-MaxRIS

In this subsection, we give the following theorem.

Theorem 5 *For every fixed constant $r \geq 0$, r-MaxRIS is solvable in linear time for graphs with bounded treewidth.*

As a proof of Theorem 5, we give such an algorithm. Indeed, we give a linear-time algorithm which simply computes $\mathrm{OPT}_{\mathsf{RIS}}(G)$ for a given graph G; it is easy to modify our algorithm so that it actually finds an r-regular induced subgraph with the maximum number $\mathrm{OPT}_{\mathsf{RIS}}(G)$ of vertices.

Main ideas.

We first give our main ideas. Let G be a graph whose treewidth is bounded by a fixed constant k, and let $\langle \{X_i \mid i \in V_T\}, T \rangle$ be a nice tree-decomposition of G. Consider an arbitrary r-regular induced subgraph F of G, and consider the subgraph F_i of F which is induced by the vertices in $V(F) \cap V(G_i)$ for a node $i \in V_T$. Then, there are the following two cases (a) and (b) to consider.

Case (a): $V(F) \cap X_i = \emptyset$. (See Fig. 3.13 as an example for $r = 2$.)

In this case, Proposition 2 implies that F_i is either empty or an r-regular induced subgraph of G_i. Note that, in the latter case, $F_i = F$ does not necessarily hold, but F_i consists of

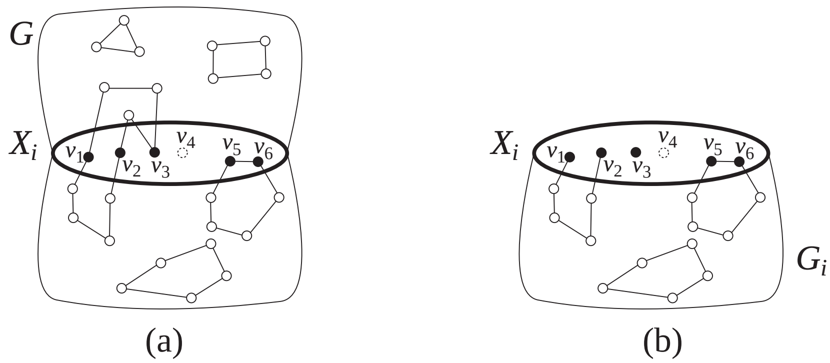


Fig. 3.14 (a) A 2-regular induced subgraph F of a graph G such that $V(F) \cap X_i \neq \emptyset$, and (b) the (K, ϕ)-subgraph F_i of G_i, where $X_i = \{v_1, v_2, \ldots, v_6\}$, $K = \{v_1, v_2, v_3, v_5, v_6\}$, $\phi(v_1) = \phi(v_2) = 1$, $\phi(v_3) = 0$ and $\phi(v_5) = \phi(v_6) = 2$.

connected components that are contained in F.

Case (b): $V(F) \cap X_i \neq \emptyset$. (See Fig. 3.14 as an example for $r = 2$.)

In this case, each connected component in F_i is not necessarily r-regular if it contains a vertex in X_i, since some vertices in X_i will be joined with vertices in $G \setminus G_i$. (See the vertices v_1, v_2, v_3 in Fig. 3.14.) On the other hand, Proposition 2 implies that every vertex in $V(F_i) \setminus X_i$ must be of degree exactly r. Note that Case (b) includes the case where both $F_i = F$ and $V(F) \cap X_i \neq \emptyset$ hold.

Motivated by Cases (a) and (b) above, we characterize induced subgraphs of G_i with respect to the degree (regularity) property on the vertices in X_i. For a node $i \in V_T$, let $K \subseteq X_i$ and let $\phi : K \to \{0, 1, \ldots, r\}$; as we will describe later, the set K will represent the vertices in X_i that are contained in an induced subgraph of G_i, and ϕ will maintain the degree property on K. We call such a pair (K, ϕ) a *pair for* X_i. Then, an induced subgraph F' of G_i is called a (K, ϕ)-*subgraph of* G_i if the following two conditions (i) and (ii) hold:

(i) $d(F', v) = r$ for every vertex v in $V(F') \setminus X_i$; and

(ii) $V(F') \cap X_i = K$, and $d(F', v) = \phi(v)$ for each vertex $v \in K$.

For the sake of convenience, we say that an empty graph (containing no vertex) is an $(\emptyset, \phi)$-subgraph of G_i. Then, an $(\emptyset, \phi)$-subgraph F' of G_i is either empty or an r-regular induced subgraph of G_i containing no vertex in X_i. Therefore, the pairs (K, ϕ) for X_i correspond to Case (a) above if $K = \emptyset$. Clearly, the following lemma holds.

Lemma 2 *A (K, ϕ)-subgraph F' of G_i is an r-regular induced subgraph of G_i if and only if $K = \emptyset$ or $\phi(v) = r$ for all vertices $v \in K$.*

We now define a value $f(i; K, \phi)$ for a node $i \in V_T$ and a pair (K, ϕ) for X_i, as follows:

$$f(i; K, \phi) = \max\{|S| : S \subseteq V(G_i) \text{ and } G[S] \text{ is a } (K, \phi)\text{-subgraph of } G_i\}.$$

If G_i has no (K, ϕ)-subgraph, then we let $f(i; K, \phi) = -\infty$. Our algorithm computes $f(i; K, \phi)$ for each node $i \in V_T$ and all pairs (K, ϕ) for X_i, from the leaves of T to the root of T, by means of dynamic programming. Then, since $G_0 = G$ for the root 0 of T, by Lemma 2 one can compute $\mathrm{OPT}_{\mathsf{RIS}}(G)$ for a given graph G, as follows:

$$\mathrm{OPT}_{\mathsf{RIS}}(G) = \max f(0; K, \phi), \tag{3.1}$$

where the maximum above is taken over all pairs (K, ϕ) for X_0 such that $K = \emptyset$ or $\phi(v) = r$ for all vertices $v \in K$.

Algorithm and its running time.

We first estimate the number of all pairs (K, ϕ) for each bag X_i. Recall that a given graph G is of treewidth bounded by a fixed constant k, and hence each bag X_i of T contains at most $k + 1$ vertices. Since $K \subseteq X_i$ and $\phi : K \to \{0, 1, \ldots, r\}$, the number of all pairs (K, ϕ) for X_i can be bounded by

$$\sum_{p=0}^{k+1} \binom{k+1}{p} \cdot (r+1)^p \leq 2^{k+1} \cdot (r+1)^{k+1} = O(1). \tag{3.2}$$

Notice that this is a single exponential with respect to k, as we have discussed in Introduction.

We now explain how to compute $f(i; K, \phi)$ for each node $i \in V_T$ and all pairs (K, ϕ) for X_i, from the leaves of T to the root of T.

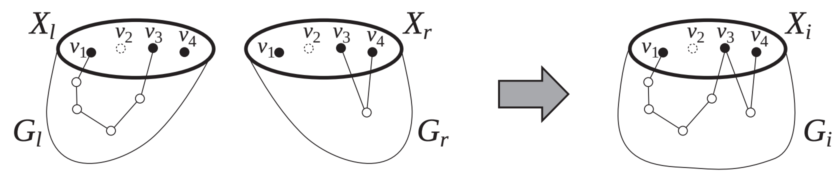


Fig. 3.15 Case (1): a (K, ϕ)-subgraph of G_i with $K = \{v_1, v_3, v_4\}$ and $\phi(v_1) = \phi(v_4) = 1$, $\phi(v_3) = 2$ which is obtained by merging a (K_l, ϕ_l)-subgraph of G_l with a (K_r, ϕ_r)-subgraph of G_r, where $K_l = K_r = K = \{v_1, v_3, v_4\}$ and $\phi_l(v_1) = \phi_l(v_3) = 1$, $\phi_l(v_4) = 0$, $\phi_r(v_1) = 0$, $\phi_r(v_3) = \phi_r(v_4) = 1$.

[The node i is a leaf of T]

For each leaf i of T, a simple brute-force algorithm can compute $f(i; K, \phi)$ for each pair (K, ϕ) for X_i. Since $G_i = G[X_i]$ contains at most $k + 1$ vertices, the number of induced subgraphs of G_i is 2^{k+1}. Therefore, this brute-force algorithm takes time $O(1)$ for each leaf i of T and all pairs (K, ϕ) for X_i. There are $O(n)$ leaves in T, and hence $f(i; K, \phi)$ can be computed in linear time for all leaves $i \in V_T$ and all pairs (K, ϕ) for X_i.

[The node i is an internal node of T]

We then compute $f(i; K, \phi)$ for each internal node i of T and each pair (K, ϕ) for X_i. Since $\langle \{X_i \mid i \in V_T\}, T \rangle$ is a nice tree-decomposition of G, there are three cases to consider, that is, i has two children, is a forget node, and is an introduce node.

Case (1): The node i has two children l and r. (See Fig. 3.15 as an example for $r = 2$.)

In this case, $X_i = X_l = X_r$. By Proposition 2 there is no edge joining a vertex in $G_l \setminus X_l$ and one in $G_r \setminus X_r$. Then, for a pair (K, ϕ) for X_i, a (K, ϕ)-subgraph of G_i can be obtained by merging a (K_l, ϕ_l)-subgraph of G_l with a (K_r, ϕ_r)-subgraph of G_r such that $K_l = K_r = K$ and $\phi_l(u) + \phi_r(u) = \phi(u)$ for all vertices $u \in K$. Therefore, we have

$$f(i; K, \phi) = \max\{f(l; K_l, \phi_l) + f(r; K_r, \phi_r)\} - |K|,$$

where the maximum above is taken over all pairs (K_l, ϕ_l) for X_l and (K_r, ϕ_r) for X_r such that

(a) $K_l = K_r = K$; and

(b) $\phi_l(u) + \phi_r(u) = \phi(u)$ for all vertices $u \in K$.

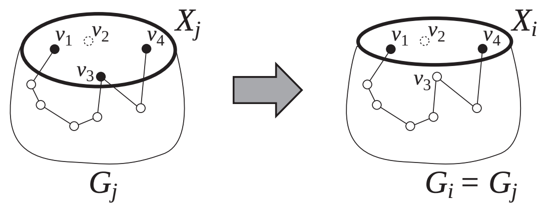


Fig. 3.16 Case (2): a (K, ϕ)-subgraph of G_i with $K = \{v_1, v_4\}$ and $\phi(v_1) = \phi(v_4) = 1$ which is obtained from a (K', ϕ')-subgraph of G_j such that $K' = \{v_1, v_3, v_4\}$ and $\phi'(v_1) = \phi'(v_4) = 1$, $\phi'(v_3) = 2$, where $v' = v_3$.

Note that, since $K_l = K_r = K$, the vertices in K are counted exactly twice in $f(l; K_l, \phi_l) + f(r; K_r, \phi_r)$.

Case (2): The node i is a forget node. (See Fig. 3.16 as an example for $r = 2$.)

In this case, the node i has exactly one child j in T such that $|X_i| = |X_j| - 1$ and $X_i \subset X_j$. Notice that $G_i = G_j$ in this case. Let v' be the vertex in $X_j \setminus X_i$. It should be noted that v' is forgotten here, and hence Proposition 2 implies that there is no edge joining a vertex in $G \setminus G_i$ and v'. Therefore, if v' is contained in an induced subgraph of G_j, then v' must be incident to exactly r vertices in $G_j = G_i$. For each pair (K, ϕ) for X_i, we thus have

$$f(i; K, \phi) = \max f(j; K', \phi'),$$

where the maximum above is taken over all pairs (K', ϕ') for X_j such that

(a) $K' \setminus \{v'\} = K$;

(b) $\phi'(u) = \phi(u)$ for all vertices $u \in K' \setminus \{v'\}$; and

(c) $\phi'(v') = r$ if $v' \in K'$.

Case (3): The node i is an introduce node. (See Fig. 3.17 as an example for $r = 2$.)

In this case, the node i has exactly one child j in T such that $|X_i| = |X_j| + 1$ and $X_i \supset X_j$. Let v' be the vertex in $X_i \setminus X_j$. Since v' is introduced by X_i, every edge in G_i incident to v' is contained in X_i, that is, $N(G_i, v') \subseteq X_i$. Then, for each pair (K, ϕ) for X_i such that $v' \notin K$, we have

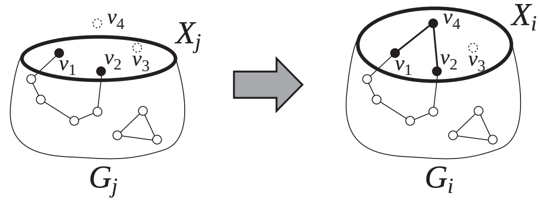


Fig. 3.17 Case (3): a (K, ϕ)-subgraph of G_i with $K = \{v_1, v_2, v_4\}$ and $\phi(v_1) = \phi(v_2) = \phi(v_4) = 2$ which is obtained from a (K', ϕ')-subgraph of G_j such that $K' = \{v_1, v_2\}$ and $\phi'(v_1) = \phi'(v_2) = 1$, where $v' = v_4$.

$$f(i; K, \phi) = f(j; K, \phi).$$

On the other hand, for each pair (K, ϕ) for X_i such that $v' \in K$, we let

$$f(i; K, \phi) = -\infty$$

if $\phi(v') \neq d(G[K], v')$; otherwise

$$f(i; K, \phi) = 1 + \max f(j; K', \phi'),$$

where the maximum above is taken over all pairs (K', ϕ') for X_j such that

(a) $K' = K \setminus \{v'\}$;

(b) for each vertex $u \in K'$,

$$\phi'(u) = \begin{cases} \phi(u) - 1 & \text{if } u \in N(G_i, v'); \\ \phi(u) & \text{otherwise.} \end{cases}$$

Remember that both r and k are assumed to be fixed constants, and that the number of all pairs (K, ϕ) for each bag X_i is $O(1)$. Therefore, all the update formulas in Cases (1)–(3) above can be computed in time $O(1)$ for all pairs (K, ϕ) for X_i.

Since T has $O(n)$ nodes, the values $f(0; K, \phi)$ can be computed in linear time for all pairs (K, ϕ) for the root 0 of T. By Eq. (3.1) the optimal value $\mathrm{OPT}_{\mathsf{RIS}}(G)$ can be computed in time $O(1)$ from the values $f(0; K, \phi)$. In this way, our algorithm runs in linear time.

This completes the proof of Theorem 5. $\square$

3.5.3 Algorithm for r-MaxRICS

In this subsection, we give the following theorem.

Theorem 6 *For every fixed constant $r \geq 0$, r-MaxRICS is solvable in linear time for graphs with bounded treewidth.*

Our algorithm for r-MaxRICS is almost the same as one for r-MaxRIS, but we take the connectivity property into account. Let G be a graph whose treewidth is bounded by a fixed constant k, and let $\langle \{X_i \mid i \in V_T\}, T \rangle$ be a nice tree-decomposition of G. For a node $i \in V_T$, let $K \subseteq X_i$, and let $\phi : K \to \{0, 1, \ldots, r\}$, $\pi : K \to \{0, 1, \ldots, k\}$; π will maintain the connectivity property on K. We call such a triple (K, ϕ, π) a *triple for X_i*. Then, an induced subgraph F' of G_i, which is not necessarily connected, is called a *(K, ϕ, π)-subgraph of G_i* if the following three conditions hold (see also Fig. 3.18 as an example for $r = 2$):

(i) $d(F', v) = r$ for every vertex v in $V(F') \setminus X_i$;

(ii) $V(F') \cap X_i = K$, and $d(F', v) = \phi(v)$ for each vertex $v \in K$; and

(iii) if $K = \emptyset$, then F' is an empty graph or consists of exactly one connected component (having no vertex in X_i); otherwise

   (a) each connected component in F' contains at least one vertex in K;

   (b) two vertices $v, w \in K$ are contained in the same connected component in F' if and only if $\pi(v) = \pi(w)$.

Notice that the condition (iii) above maintains the connectivity property: Condition (iii)-(a) ensures that the distinct components in F' can be merged into a single connected component (recall Proposition 2); and by Condition (iii)-(b) the value $\pi(v)$ identifies the connected component containing v. Note that, since each bag X_i contains at most $k + 1$ vertices, there are at most $k + 1$ different connected components in F'. Then, the following lemma clearly holds.

Lemma 3 *A (K, ϕ, π)-subgraph F' of G_i is an r-regular induced connected subgraph of G_i if $K = \emptyset$, or $\phi(v) = r$ for all vertices $v \in K$ and $|\{\pi(v) : v \in K\}| = 1$.*

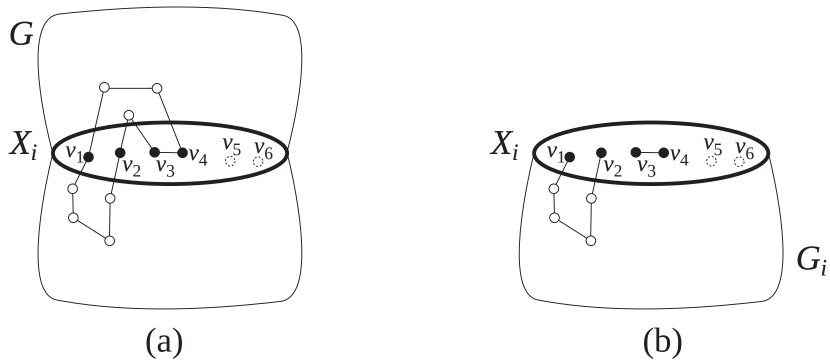


Fig. 3.18 (a) A 2-regular induced connected subgraph F of a graph G, and (b) the (K, ϕ, π)-subgraph F' of G_i, where $X_i = \{v_1, v_2, \ldots, v_6\}$, $K = \{v_1, v_2, v_3, v_4\}$, $\phi(v_1) = \phi(v_2) = \phi(v_3) = \phi(v_4) = 1$, $\pi(v_1) = \pi(v_2)$ and $\pi(v_3) = \pi(v_4)$ with $\pi(v_1) \neq \pi(v_3)$.

As the counterpart of $f(i; K, \phi)$ for r-MaxRIS, we define a value $g(i; K, \phi, \pi)$ for a node $i \in V_T$ and a triple (K, ϕ, π) for X_i, as follows:

$$g(i; K, \phi, \pi) = \max\{|S| : S \subseteq V(G_i) \text{ and } G[S] \text{ is a } (K, \phi, \pi)\text{-subgraph of } G_i\}.$$

If G_i has no (K, ϕ, π)-subgraph, then we let $g(i; K, \phi, \pi) = -\infty$. Similarly as in Section 3.5.2, our algorithm computes $g(i; K, \phi, \pi)$ for each node $i \in V_T$ and all triples (K, ϕ, π) for X_i, from the leaves of T to the root of T, by means of dynamic programming. Then, since $G_0 = G$ for the root 0 of T, by Lemma 3 one can compute $\mathrm{OPT}_{\mathsf{RICS}}(G)$ for a given graph G, as follows:

$$\mathrm{OPT}_{\mathsf{RICS}}(G) = \max g(0; K, \phi, \pi),$$

where the maximum above is taken over all triples (K, ϕ, π) for X_0 such that either $K = \emptyset$, or $\phi(v) = r$ for all vertices $v \in K$ and $|\{\pi(v) : v \in K\}| = 1$.

Note that the number of all triples (K, ϕ, π) for each bag X_i can be bounded by

$$\sum_{p=0}^{k+1} \binom{k+1}{p} \cdot (r+1)^p \cdot (k+1)^p \leq 2^{k+1} \cdot (r+1)^{k+1} \cdot (k+1)^{k+1} = O(1). \cdots\cdots\cdots\cdots (3.3)$$

Therefore, by similar arguments as in Section 3.5.2, we can conclude that our modified algorithm solves r-MaxRICS in linear time for graphs with bounded treewidth. □

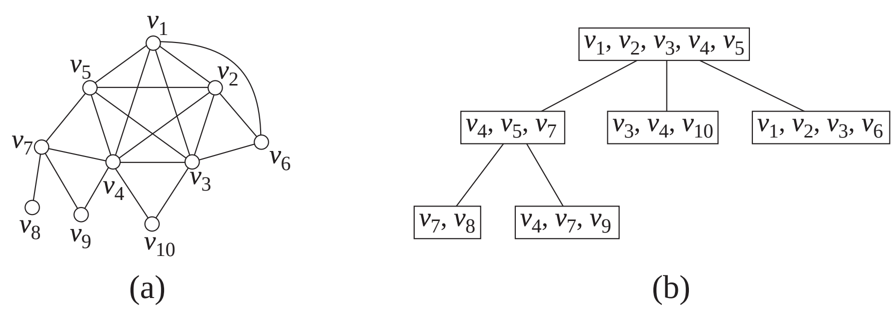


Fig. 3.19 (a) Chordal graph G and (b) its clique tree T.

3.6 Chordal Graphs

In this section, we consider the problems restricted to chordal graphs. A graph G is *chordal* if every cycle in G of length at least four has at least one chord, which is an edge joining non-adjacent vertices in the cycle [8]. (See Fig. 3.19(a) as an example.)

3.6.1 Definitions and key lemma

Let $\mathcal{K}_G$ be the set of all maximal cliques in a graph G, and let $\mathcal{K}_v \subseteq \mathcal{K}_G$ be the set of all maximal cliques that contain a vertex $v \in V(G)$. It is known that G is chordal if and only if there exists a tree $T = (\mathcal{K}_G, E)$ such that each node of T corresponds to a maximal clique in $\mathcal{K}_G$ and the induced subtree $T[\mathcal{K}_v]$ is connected for every vertex $v \in V(G)$ [5]. (See Fig. 3.19 as an example.) Such a tree is called a *clique tree* of G, and it can be constructed in linear time [5]. Indeed, a clique tree of a chordal graph G is a tree-decomposition of G. Therefore, we call a clique in $\mathcal{K}_G$ also a *node* of T, and refer to the subgraph G_C corresponding to a node C defined as in Section 3.5.1. For the sake of notational convenience, each node C of T simply indicates the vertex set $V(C)$; we represent the clique corresponding to C by $G[C]$. For a node $C \in \mathcal{K}_G$, we denote by $p(C)$ the parent of C in T; let $p(C_0) = \emptyset$ for the root node C_0 of T.

We now give the key lemma to design our algorithms.

Lemma 4 *Every regular induced subgraph of a chordal graph is a clique.*

proof 4 *Assume for a contradiction that a chordal graph G has a regular induced subgraph G' which is not a clique. Since G' is an induced subgraph of a chordal graph, G' is also chordal and hence there is a clique tree $T' = (\mathcal{K}_{G'}, E')$ for G'. In addition, since G' is not a clique, T' has at least two nodes. Consider any leaf node C in T', and let $P = p(C)$. Recall that both of C and P correspond to different maximal cliques in G'. We thus have $C \setminus P \neq \emptyset$ and $P \setminus C \neq \emptyset$. Furthermore, $P \cap C \neq \emptyset$ since C and P are adjacent in T.*

Let $v_c \in C \setminus P$ and $v_{pc} \in P \cap C$. Since v_c belongs only to the node C and $G'[C]$ forms a clique, we have $d(G', v_c) = |C| - 1$. On the other hand, since v_{pc} belongs to (at least) two cliques $G'[C]$ and $G'[P]$, its degree in G' is

$$d(G', v_{pc}) \geq |C \setminus P| + |P \setminus C| + (|C \cap P| - 1) = |C| + |P \setminus C| - 1 \geq |C|,$$

where the last inequality comes from the fact that $P \setminus C \neq \emptyset$, i.e., *$|P \setminus C| \geq 1$. Therefore, we obtain $d(G', v_c) = |C| - 1$ and $d(G', v_{pc}) \geq |C|$, which contradicts the assumption that G' is regular.* □

3.6.2 Algorithm for r-MaxRICS

Based on Lemma 4, we give the following theorem. Note that the degree constraint r is not necessarily a fixed constant.

Theorem 7 *For every integer $r \geq 0$, r-*MaxRICS *is solvable in polynomial time for chordal graphs.*

proof 5 *Lemma 4 implies that r-*MaxRICS *for a chordal graph G is equivalent to finding a clique of size $r + 1$ in G, which can be done in polynomial time by utilizing a polynomial-time algorithm to find a maximum clique in chordal graphs [18]: Find a maximum clique of G; if the maximum clique is of size at least $r + 1$, then* $\mathrm{OPT}_{\mathsf{RICS}}(G) = r + 1$*; otherwise* $\mathrm{OPT}_{\mathsf{RICS}}(G) = 0$. □

3.6.3 Algorithm for r-MaxRIS

In this subsection, we give the following theorem.

Theorem 8 *For every integer $r \geq 0$, r-MaxRIS can be solved in time $O(n^2)$ for chordal graphs, where n is the number of vertices in a given graph.*

As a proof of Theorem 8, we give such an algorithm. Similarly as for r-MaxRICS, Lemma 4 implies that r-MaxRIS for a chordal graph G is equivalent to finding the maximum number of "independent" cliques of size $r + 1$ in G. From now on, we call a clique of size exactly $r + 1$ an $(r + 1)$-*clique*. We say that $(r + 1)$-cliques in G are *independent* if no two vertices in different $(r + 1)$-cliques are adjacent in G. For an induced subgraph G' of a chordal graph G, we denote by $\#_{r+1}(G')$ the maximum number of independent $(r+1)$-cliques in G'. Then,

$$\mathrm{OPT}_{\mathsf{RIS}}(G) = (r + 1) \cdot \#_{r+1}(G).$$

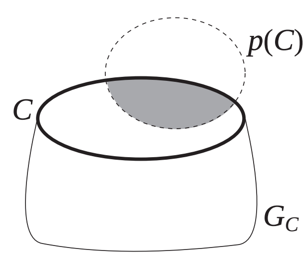


Fig. 3.20 Subgraph G_C and the parent $p(C)$ for a node C of a clique tree T.

Main idea and our algorithm.

Let T be a clique tree for a given chordal graph G. Since each node of T corresponds to a maximal clique of G, for any $(r + 1)$-clique K there exists at least one node C of T such that $G[C]$ contains K. Therefore, roughly speaking, our algorithm determines whether the vertices in a node of T can be selected as an $(r + 1)$-clique or not, by traversing the nodes from the leaves of T to the root of T, so that the number of independent $(r + 1)$-cliques in G is maximized.

Note that, however, there are several vertices that are contained in more than one nodes of T, and hence we need to be careful for keeping the independency of $(r + 1)$-cliques when we select $(r + 1)$-cliques. Such a vertex must be in $C \cap p(C)$ for every two adjacent nodes C and $p(C)$ of T. (See Fig. 3.20.) Therefore, we can select one $(r + 1)$-clique from $G_C \setminus p(C)$ without collision with any $(r+1)$-clique in $G \setminus G_C$. (This claim will be proven formally later.) We label a node C of T as *small* if and only if $G_C \setminus p(C)$ contains no $(r + 1)$-clique; namely, the subgraph $G_C \setminus p(C)$ is too *small* to select an $(r + 1)$-clique. It should be noted that, even if C is labeled with small, there may exist an $(r + 1)$-clique in G_C which must contain some vertices in $C \cap p(C)$.

We describe our algorithm for r-MaxRIS below. For the sake of convenience, we regard that each leaf of a clique tree has one dummy child which is labeled with small; then, Step 2 will be executed for each unlabeled original leaf node. Remember that $p(C_0) = \emptyset$ for the root node C_0 of a clique tree.

Initialization. $S := \emptyset$, $G' := G$ and construct a clique tree T' for G'.

Step 1. If G' is empty or all nodes of T' are labeled with small, then output S.

Step 2. Pick any unlabeled node C of T' whose all children are labeled with small.

- **(a)** If $G_C \setminus p(C)$ contains an $(r + 1)$-clique, then add its $r + 1$ vertices to S. Set $G' := G' \setminus G_C$, and modify the clique tree for the new graph G'. Then, goto Step 1.
- **(b)** Otherwise label C as small, and goto Step 1.

Note that, if Step 2(a) results in a disconnected chordal graph G', then we apply our algorithm to each connected component in G'. This algorithm runs in time $O(n^2)$, where $n = |V(G)|$, because

(1) a clique tree T has $O(n)$ nodes;

(2) each step can be done in time $O(n)$; and

(3) one execution of Step 2 deletes at least one node, or labels one node.

To complete the proof of Theorem 8, we now show that our algorithm above correctly solves r-MaxRIS for chordal graphs. Notice that $(r+1)$-cliques are selected only in Step 2(a), and hence it suffices to show the following lemma.

Lemma 5 *For an unlabeled node C of T', suppose that all children of C in T' are labeled with small, and that $G_C \setminus p(C)$ contains an $(r+1)$-clique. Then,*

$$\#_{r+1}(G') = \#_{r+1}(G' \setminus G_C) + 1.$$

Proof of Lemma 5.

We first show an important property on clique trees. For a vertex subset V' of a connected graph G, we say that V' *separates two vertices u and v* if u and v belong to different connected components in $G \setminus V'$.

Lemma 6 ([5]) *For every two adjacent nodes C and $p(C)$ in T', the set $C \cap p(C)$ separates any vertex in $G_C \setminus p(C)$ and any vertex in $G' \setminus G_C$.*

Lemma 6 implies that any $(r+1)$-clique in $G_C \setminus p(C)$ is independent from any $(r+1)$-clique in $G' \setminus G_C$, and *vice versa*. (See also Fig. 3.20.) Therefore, if $G_C \setminus p(C)$ contains an $(r+1)$-clique, then we have

$$\#_{r+1}(G') \geq \#_{r+1}(G' \setminus G_C) + 1.$$

To complete the proof of Lemma 5, we thus verify $\#_{r+1}(G') \leq \#_{r+1}(G' \setminus G_C)+1$ in Lemma 8. We now show the following auxiliary lemma.

Lemma 7 *For an unlabeled node C of T', suppose that all children of C in T' are labeled with small. Let S be an arbitrary subset of $V(G')$ such that $G'[S]$ forms independent $(r+1)$-cliques. If S contains an $(r+1)$-clique K such that $V(K) \cap V(G_C) \neq \emptyset$, then no other vertex in $V(G_C)$ is contained in S, that is, $(S \setminus V(K)) \cap V(G_C) = \emptyset$.*

proof 6 *Since each child C_i of C is labeled with small, the subgraph $G_{C_i} \setminus p(C_i) = G_{C_i} \setminus C$ contains no $(r+1)$-clique. Furthermore, by Lemma 6 no vertex in $G_{C_i} \setminus C$ is connected to a* *vertex in $G' \setminus G_{C_i}$. Therefore, if S contains an $(r+1)$-clique K such that $V(K) \cap V(G_C) \neq \emptyset$, then K must contain at least one vertex in C.*

Suppose for a contradiction that $(S \setminus V(K)) \cap V(G_C) \neq \emptyset$. Then, there exists another $(r+1)$-clique K' such that $K' \neq K$ and $V(K') \cap V(G_C) \neq \emptyset$. The same argument implies that K' contains at least one vertex in C. However, since $G'[C]$ is a (maximal) clique, this contradicts the independency of $(r+1)$-cliques in $G'[S]$. □

We finally give the following lemma, and complete the proof of Lemma 5.

Lemma 8 *For an unlabeled node C of T', suppose that all children of C in T' are labeled with small. Then, $\#_{r+1}(G') \leq \#_{r+1}(G' \setminus G_C) + 1$.*

proof 7 *Let $X^* \subseteq V(G')$ be an arbitrary optimal solution for G'. Then,*

$$\#_{r+1}(G') = \#_{r+1}(G'[X^*]). \cdots\cdots (3.4)$$

By Lemma 7 there exists at most one clique K in X^* which contains a vertex in G_C. Let $G^{'-}$ be the induced subgraph of $G' \setminus G_C$ which is obtained from G' by deleting all vertices in G_C and in K (if there exists). Then, $X^* \setminus V(K)$ forms independent $(r+1)$-cliques in $G^{'-}$, and hence we have

$$\#_{r+1}(G'[X^* \setminus V(K)]) \leq \#_{r+1}(G^{'-}). \cdots\cdots (3.5)$$

By Eqs. (3.4) and (3.5) we have

$$\#_{r+1}(G') = \#_{r+1}(G'[X^*]) = \#_{r+1}(G'[X^* \setminus V(K)]) + 1 \leq \#_{r+1}(G^{'-}) + 1. \cdots\cdots (3.6)$$

Since $G^{'-}$ is an induced subgraph of $G' \setminus G_C$, we have $\#_{r+1}(G^{'-}) \leq \#_{r+1}(G' \setminus G_C)$. Therefore, by Eq. (3.6) we have $\#_{r+1}(G') \leq \#_{r+1}(G^{'-}) + 1 \leq \#_{r+1}(G' \setminus G_C) + 1$. □

3.7 Concluding Remarks

In this chapter, we studied the complexity statuses of the r-MaxRIS and r-MaxRICS problems from the viewpoint of graph classes, and analyzed which graph property makes the problems tractable/intractable.

We remark that both of our algorithms for graphs with bounded treewidth run in polynomial time even if the degree constraint r is not a fixed constant; see Eqs. (3.2) and (3.3). Furthermore, these algorithms can be easily modified so that they solve more general problems, defined as follows: Given a bounded treewidth graph G and two integers l and u with $l \leq u$, we wish to find a maximum vertex-subset S of G such that every vertex in $G[S]$ is of degree at least l and at most u; as a variant, we may consider the problem which requires $G[S]$ to be connected. Then, these two problems are generalization of r-MaxRIS and r-MaxRICS; consider the case where $l = u = r$.

Chapter 4

Distance d independent set

4.1 Introduction

Recall from Chapter 1 that one of the most important and most investigated computational problems in theoretical computer science and combinatorial optimization is the Independent Set problem (IS for short) because of its many applications in scheduling, computer vision, pattern recognition, coding theory, map labeling, computational biology, and some other fields. The input of IS is an unweighted graph $G = (V, E)$ and a positive integer $k \leq |V|$. An *independent set* of G is a subset $S \subseteq V$ of vertices such that, for all $u, v \in S$, the edge $\{u, v\}$ is not in E. IS asks whether G contains an independent set S having $|S| \geq k$. IS is among the first problems ever to be shown to be $\mathcal{NP}$-complete, and has been used as a starting point for proving the $\mathcal{NP}$-completeness of other problems [17]. Moreover, it is well known that IS remains $\mathcal{NP}$-complete even for substantially restricted graph classes such as cubic planar graphs [16], triangle-free graphs [34], and graphs with large girth [32].

In this chapter, we consider a generalization of IS, named the Distance-d Independent Set problem (DdIS for short). A distance-d independent set for an integer $d \geq 2$ in an unweighted graph $G = (V, E)$ is a subset $S \subseteq V$ of vertices such that for any pair of vertices $u, v \in S$, the distance between u and v is at least d in G. For a fixed constant $d \geq 2$, DdIS considered in this chapter is formulated as the following class of problems [1]:

> DISTANCE-d INDEPENDENT SET (DdIS)
>
> **Input:** An unweighted graph $G = (V, E)$ and a positive integer $k \leq |V|$.
>
> **Question:** Does G contain a distance-d independent set of size k or more?

The maximization version of DdIS can be also defined:

> MAXIMUM DISTANCE-d INDEPENDENT SET (MaxDdIS)
>
> **Input:** An unweighted graph $G = (V, E)$.
>
> **Output:** A distance-d independent set of the maximum size.

The problem parameterized by the solution size k is as follows:

> PARAMETERIZED DISTANCE-d INDEPENDENT SET (ParaDdIS(k))
>
> **Input:** An unweighted graph $G = (V, E)$.
>
> **Parameter** A positive integer $k \leq |V|$.
>
> **Question:** Does G contain a distance-d independent set of size k or more?

It is important to note that D2IS is identical to the original IS, and DdIS is equivalent to IS on the $(d-1)$th power graph G^{d-1} of the input graph G as pointed out in [1].

Even when $d = 2$, DdIS (i.e., D2IS) is $\mathcal{NP}$-complete, and thus it would be easy to show that DdIS is $\mathcal{NP}$-complete in general. Fortunately, however, it is known that if the input graph is restricted to, for example, bipartite graphs [22], chordal graphs [18], circular-arc graphs [19], comparability graphs [20], and many other classes [31, 29, 7], then D2IS admits polynomial-time algorithms. Furthermore, Agnarsson, Damaschke, Halldórsson [1] show the following tractability of DdIS by using the closure property under taking power [14, 15, 35]:

Fact 2 ([1]) *Let n denote the number of vertices in the input graph G. Then, for every integer $d \geq 2$,* DdIS *is solvable in $O(n)$ time for interval graphs, in $O(n(\log\log n + \log d))$ time for trapezoid graphs, and in $O(n)$ time for circular-arc graphs.*

This tractability suggests that if we restrict the set of instances to, for example, subclasses of bipartite graphs and chordal graphs, then DdIS for a fixed $d \geq 3$ might be also solvable efficiently. On the other hand, however, we have a "negative" fact that if G is planar/bipartite, then the $(d-1)$th power graph G^{d-1} is not necessarily planar/bipartite. From those points of view, this chapter investigates DdIS, namely, our work focuses on the computational complexity of DdIS and/or the inapproximability of MaxDdIS on (subclasses of) bipartite graphs and chordal graphs.

Our main results are summarized in the following list:

(i) For every fixed integer $d \geq 3$, DdIS is $\mathcal{NP}$-complete even for bipartite graphs.

(ii) For any $\varepsilon > 0$ and fixed integer $d \geq 3$, it is $\mathcal{NP}$-hard to approximate MaxDdIS to within a factor of $n^{1/2-\varepsilon}$ for bipartite graphs of n vertices.

(iii) For every fixed integer $d \geq 3$, ParaDdIS(k) is $\mathcal{W}[1]$-hard for bipartite graphs.

(iv) For every fixed integer $d \geq 3$, DdIS remains $\mathcal{NP}$-complete even for planar bipartite graphs of maximum degree three.

(v) For every fixed even integer $d \geq 2$, DdIS is in $\mathcal{P}$ for chordal graphs.

(vi) For every fixed odd integer $d \geq 3$, DdIS is $\mathcal{NP}$-complete for chordal graphs.

(vii) For any $\varepsilon > 0$ and fixed odd integer $d \geq 3$, it is $\mathcal{NP}$-hard to approximate MaxDdIS to within a factor of $n^{1/2-\varepsilon}$ for chordal graphs of n vertices.

(viii) For every fixed odd integer $d \geq 3$, ParaDdIS(k) is $\mathcal{W}[1]$-hard for chordal graphs.

One can see that the complexity of DdIS depends on the parity of d if the set of input graphs is restricted to chordal graphs.

The organization of the chapter is as follows: Section 4.2 is devoted to our notation and terminology. In Section 4.3 we prove the $\mathcal{NP}$-hardness, the hardness of approximation, and the $\mathcal{W}[1]$-hardness of the problem for bipartite graphs. In Section 4.4, we provide tractable and intractable cases for chordal graphs.

4.2 Notation

Let $G = (V, E)$ be an unweighted graph, where V and E denote the set of vertices and the set of edges, respectively. $V(G)$ and $E(G)$ also denote the vertex set and the edge set of G, respectively. We denote an edge with endpoints u and v by $\{u, v\}$. For a pair of vertices u and v, the length of a shortest path from u to v, i.e., the distance between u and v is denoted by $dist_G(u, v)$, and the diameter G is defined as $diam(G) = \max_{u,v \in V} dist_G(u, v)$.

A graph G_S is a subgraph of a graph G if $V(G_S) \subseteq V(G)$ and $E(G_S) \subseteq E(G)$. For a subset of vertices $U \subseteq V$, let $G[U]$ be the subgraph induced by U. For a subgraph $G_S = (V_S, E_S)$ of G, if $E_S = V_S \times V_S$, then G_S (or $G[V_S]$) and V_S are called a *clique* and a *clique set*, respectively.

For a positive integer $d \geq 1$ and a graph G, the dth power of G, denoted by $G^d = (V(G), E^d)$, is the graph formed from $V(G)$, where all pairs of vertices $u, v \in G$ such that $dist_G(u, v) \leq d$ are connected by an edge $\{u, v\}$. Note that $E(G) \subseteq E^d$, i.e., the original edges in $E(G)$ are retained.

A path of length ℓ, denoted by P_ℓ, from a vertex v_0 to a vertex v_ℓ is represented as a sequence of vertices such that $P_\ell = \langle v_0, v_1, \cdots, v_\ell \rangle$. A cycle of length ℓ, denoted by C_ℓ, is similarly written as $C_\ell = \langle v_0, v_1, \cdots, v_{\ell-1}, v_0 \rangle$. A *chord* of a path (cycle) is an edge between two vertices of the path (cycle) that is not an edge of the path (cycle).

A graph $G = (V, E)$ is *bipartite* if there is a partition of V into two disjoint independent sets V_1 and V_2 such that $V_1 \cup V_2 = V$. A *planar bipartite* graph is a bipartite graph that can be drawn in the plane without edge crossings. A graph G is *chordal* if each cycle in G of length at least four has at least one chord. A graph $G = (V, E)$ is *split* if there is a partition of V into a clique set V_1 and an independent set V_2 such that $V_1 \cap V_2 = \emptyset$ and $V_1 \cup V_2 = V$. Note that the split graphs are a subclass of the chordal graphs. A graph is *star* if it is a rooted tree of height one. See, e.g., [8], for the definitions of interval, trapezoid, circular-arc, and comparability graphs, and inclusion relations among the graph classes.

For the maximization problems, an algorithm ALG is called a σ-approximation algorithm and the approximation ratio of ALG is σ if $OPT(G)/ALG(G) \leq \sigma$ holds for every input G, where $ALG(G)$ and $OPT(G)$ are the number of vertices of obtained subsets by ALG and the number of vertices of an optimal solution, respectively.

A *parameterized problem* is a pair (Q, k) where $Q \subseteq \Sigma^*$ is a decision problem over some alphabet Σ, and $k : \Sigma^* \to \mathbf{N}$ is a *parameterization* of the problem, assigning a *parameter* to each instance of Q. An algorithm is *fixed-parameter tractable* or *fpt* if it has a running time at most $f(k) \cdot n^c$ for some computable function f and a constant c, where n is the input length and k is the parameter assigned to the input. Given two parameterized problems (Q_1, k_1) and (Q_2, k_2) over the alphabet Σ, an *fpt-reduction* from (Q_1, k_1) to (Q_2, k_2) is a function $g : \Sigma^* \to \Sigma^*$, computable by an fpt-algorithm, such that $I \in Q_1$ if and only if $g(I) \in Q_2$ and $k_2(g(I)) \leq f(k_1(I))$ for some computable function f, for every $I \in \Sigma^*$.

4.3 Bipartite Graphs

In this section we consider the class of bipartite graphs and its subclasses. As mentioned in Section 4.1, D2IS is solvable in polynomial time by using a polynomial time algorithm which finds the maximum matching in a given bipartite graph [22]. Unfortunately, however, we can show the $\mathcal{NP}$-hardness of DdIS, the hardness of approximation of MaxDdIS, and the $\mathcal{W}[1]$-hardness of ParaDdIS(k) on bipartite graphs when $d \geq 3$.

Theorem 9 *For every fixed integer $d \geq 3$,* DdIS *is $\mathcal{NP}$-complete even for bipartite graphs.*

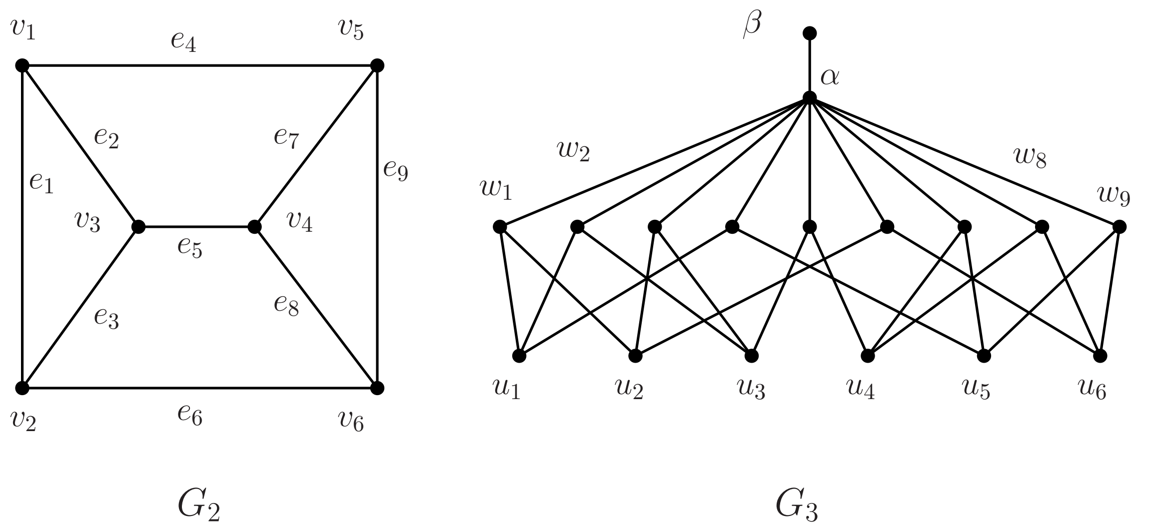


Fig. 4.1 (Left) graph G_2 of D2IS and (Right) reduced graph G_3 of D3IS from G_2.

proof 8 *We first show the $\mathcal{NP}$-completeness of* D3IS *and then one of the general* DdIS *for $d \geq 4$ in order to make the basic ideas of this proof clear. It is obvious that* DdIS *is in $\mathcal{NP}$ for every $d \geq 3$. To show that* D3IS *is $\mathcal{NP}$-hard, we reduce the $\mathcal{NP}$-hard problem* D2IS *on any general graphs to* D3IS *on bipartite graphs. That is, given a graph $G_2 = (V_2, E_2)$ of* D2IS *with n vertices, $V_2 = \{v_1, v_2, \cdots, v_n\}$, and m edges, $E_2 = \{e_1, e_2, \cdots, e_m\}$, we construct a new bipartite graph G_3 in the following way. The constructed graph G_3 consists of (i) n vertices, u_1 through u_n, each u_i of which is corresponding to $v_i \in V_2$, (ii) m vertices, w_1 through w_m, each w_i of which is corresponding to $e_i \in E_2$, and (iii) two special vertices α and β. (iv) The vertex α is connected to each vertex in $\{\beta\} \cup \{w_1, \cdots, w_m\}$, i.e., the induced graph $G[\{\alpha, \beta\} \cup \{w_1, \cdots, w_m\}]$ is star. (v) If $e_i = \{v_j, v_k\} \in E_2$, then we add two edges $\{w_i, u_j\}$ and $\{w_i, u_k\}$. Since there is a partition of V_3 into two disjoint independent sets $\{\beta, w_1, \cdots, w_m\}$ and $\{\alpha, u_1, \cdots, u_n\}$, the reduced graph G_3 must be bipartite. See Figure 4.1. For example, if the instance G_2 is the left graph, then the reduced graph G_3 is illustrated in the right graph. It is clear that this reduction can be done in polynomial time.*

For the above construction of G_3, we show that G_3 has a distance-3 independent set S_3 such that $|S_3| \geq k + 1$ if and only if G_2 has a distance-2 independent set S_2 such that $|S_2| \geq k$.

(If part) Suppose that the graph G_2 of D2IS *has the distance-2 independent set $S_2 = \{v_{1^*}, v_{2^*}, \cdots v_{k^*}\}$ in G_2, where $\{1^*, 2^*, \cdots, k^*\} \subseteq \{1, 2, \cdots, n\}$. Then, we select a subset of vertices $S_3 = \{u_{1^*}, u_{2^*}, \cdots, u_{k^*}\} \cup \{\beta\}$ of size $k+1$. Note that the distance $dist_{G_3}(\beta, u_i)$ for every i is at least three. Since the distance $dist_{G_2}(v_{i^*}, v_{j^*})$ for any pair of vertices $v_{i^*}, v_{j^*} \in S_2$ ($i^* \neq j^*$) is at least two, the shortest path from u_{i^*} to u_{j^*} contains at least two vertices in $\{w_1, w_2, \cdots, w_m\}$. This means that the distance $dist_{G_3}(u_{i^*}, u_{j^*})$ for any $i^* \neq j^*$ is at least four. Thus, the selected vertex set S_3 of size $k + 1$ is a distance-3 independent set in G_3.*

(Only-if part) Conversely, suppose that the constructed graph G_3 has the distance-3 independent set S_3 such that $|S_3| \geq k + 1$. First, take a look at the induced subgraph $G[\{\alpha, \beta\} \cup \{w_1, \cdots, w_m\}]$. Since its diameter $diam_{G_3}(G[\{\alpha, \beta\} \cup \{w_1, \cdots, w_m\}])$ is two, $|S_3 \cap V(G[\{\alpha, \beta\} \cup \{w_1, \cdots, w_m\}])| \leq 1$ holds, i.e., $|S_3 \cap \{u_1, u_2, \cdots, u_n\}| \geq k$ must be satisfied. Let $\{u_{1^*}, u_{2^*}, \cdots, u_{k^*}\}$ be a subset of k vertices in $S_3 \cap \{u_1, u_2, \cdots, u_n\}$. Then, the pairwise distance of vertices in $\{v_{1^*}, v_{2^*}, \cdots, v_{k^*}\}$ of G_2 corresponding to $\{u_{1^*}, u_{2^*}, \cdots, u_{k^*}\}$ in G_3 is surely at least 2, i.e., G_2 has a distance-2 independent set S_2 such that $|S_2| \geq k$. This completes the proof of the $\mathcal{NP}$-hardness of D3IS.

To prove the $\mathcal{NP}$-hardness of DdIS *for $d \geq 4$, we add the following two small modifications to the constructed graph G_3 in the above reduction, and construct a new bipartite graph G_d. Let $L = (d - 3) - \lceil \frac{d-1}{4} \rceil$ and let $\overline{L} = \lceil \frac{d-1}{4} \rceil$. Note that $L + \overline{L} = d - 3$. (1) The top vertex β in Figure 4.1 is replace with a simple path of length L say, $\langle \beta, \beta_1, \cdots, \beta_L \rangle$, and (2) every bottom vertex u_j is replaced with a simple path of length $\overline{L}$, say, $\langle u_j, u_{j,1}, \cdots, u_{j,\overline{L}} \rangle$ for $1 \leq j \leq n$. Then, we can again show that G_d has a distance-d independent set S_d such that $|S_d| \geq k + 1$ if and only if G_2 has a distance-2 independent set S_2 such that $|S_2| \geq k$.*

(If part for $d \geq 4$) If G_2 of D2IS *has a distance-2 independent set $S_2 = \{v_{1^*}, v_{2^*}, \cdots v_{k^*}\}$ in G_2 as before, then G_d has a subset of vertices $S_d = \{u_{1^*,\overline{L}}, u_{2^*,\overline{L}}, \cdots, u_{k^*,\overline{L}}\} \cup \{\beta_L\}$ of size $k + 1$, which must be a distance-d independent set since $dist_{G_d}(\beta_L, u_{i^*,\overline{L}}) = L + \overline{L} + 3 = (d-3) + 3 = d$ and $dist_{G_d}(u_{i^*,\overline{L}}, u_{j^*,\overline{L}}) = 4(\overline{L} + 1) = 4\lceil \frac{d-1}{4} \rceil + 4 \geq d$ for any $i^* \neq j^*$.*

(Only-if part for $d \geq 4$) Conversely, suppose that the constructed graph G_d has the

distance-d independent set S_d such that $|S_d| \geq k+1$. Similarly to the case of $d = 3$, since $diam_{G_d}(G[\{\alpha, \beta, \beta_1, \cdots, \beta_L\} \cup \{w_1, \cdots, w_m\}]) \leq d$, which means that $|S_d \cap (\{\alpha, \beta, \beta_1, \cdots, \beta_L\} \cup \{w_1, \cdots, w_m\})| \leq 1$ holds, $|S_d \cap \{u_1, u_{1,1}, \cdots, u_{1,\overline{L}}, u_2, u_{2,1}, \cdots, u_{2,\overline{L}}, \cdots, u_n, \cdots, u_{n,\overline{L}}\}| \geq k$ must be satisfied. Now we can assume that (at least) those k vertices in S_d are in the set of bottom vertices $\{u_{1,\overline{L}}, u_{2,\overline{L}}, \cdots, u_{n,\overline{L}}\}$, because $|S_d \cup \{u_{j,\overline{L}}\} \setminus \{u_{j,L'}\}| \geq |S_d|$ even if $u_{j,L'} \in S_d$ for $L' < L$. Let $\{u_{1^*,\overline{L}}, u_{2^*,\overline{L}}, \cdots, u_{k^*,\overline{L}}\}$ be a subset of k vertices in $S_d \cap \{u_{1,\overline{L}}, \cdots, u_{n,\overline{L}}\}$. Then, the pairwise distance of vertices in $\{v_{1^*}, v_{2^*}, \cdots, v_{k^*}\}$ of G_2 corresponding to $\{u_{1^*,\overline{L}}, \cdots, u_{k^*,\overline{L}}\}$ in G_d is surely at least 2, i.e., G_2 has a distance-d independent set S_2 such that $|S_2| \geq k$. This completes the proof of the theorem. □

Next, we consider the maximization version Max*Dd*IS of D*d*IS, which asks for a distance-d independent set of the maximum size in an input graph G. Since MaxD2IS is equivalent to Maximum Independent Set, it cannot be approximated within a factor of $n^{1-\varepsilon}$ [37]. In the following, we will show that the above reduction can preserve the approximation-gap and thus gives us the following inapproximability of Max*Dd*IS for $d \geq 3$.

Corellary 4 *For any $\varepsilon > 0$ and a fixed integer $d \geq 3$, it is $\mathcal{NP}$-hard to approximate* MaxD*d*IS *to within a factor of $n^{1/2-\varepsilon}$ for bipartite graphs of n vertices.*

proof 9 *Let $OPT(G_2)$ denote the number of vertices of an optimal solution for an n-vertex input graph G_2 of* MaxD2IS. *Let $OPT'(G_d)$ denote the number of vertices of an optimal solution for a ν-vertex input bipartite graph G_d of* MaxD*d*IS *for a fixed $d \geq 3$. Let $g(n)$ be a parameter function of the instance G_2 of* D2IS. *Note that the reduction described in the proof of Theorem 9 is the following gap-preserving reduction: (1) If $OPT(G_2) \geq g(n)$, then $OPT'(G_d) \geq g(n) + 1$, and (2) if $OPT(G_2) < \frac{g(n)}{n^{1-\varepsilon}}$ for a positive constant ε, then $OPT'(G_d) < \frac{g(n)}{n^{1-\varepsilon}} + 1$.*

The constructed graph G_d has at most $n \times \frac{n}{4}$ vertices labeled "u", $m \leq \frac{n^2}{2}$ vertices labeled "w", at most n vertices labeled "β", and one vertex α, i.e., $|V(G_d)| = \nu = O(n^2)$. Hence the approximation-gap is $n^{1-\varepsilon} = \Theta(\nu^{1/2-\varepsilon})$ for any $\varepsilon > 0$. By renaming ν to n, we obtain the $n^{1/2-\varepsilon}$-inapproximability of MaxD*d*IS *on bipartite graphs of n vertices.* □

Also, the reduction in the proof of Theorem 9 shows the following *fixed-parameter intractability* of ParaDdIS(k):

Corellary 5 *For every fixed integer $d \geq 3$,* ParaDdIS(k) *is $\mathcal{W}[1]$-hard for bipartite graphs.*

proof 10 *It is known [12] that* ParaD2IS(k) *on general graphs is $\mathcal{W}[1]$-hard. Let (G_2, k) and (G_d, k') be the instances of* ParaD2IS(k) *and* ParaDdIS(k') *on bipartite graphs, respectively. Then, the reduction in the proof of Theorem 9 is the fpt-reduction such that (i) $k' \leq k + 1$, and (ii) (G_2, k) is a* yes-*instance of* ParaD2IS(k) *if and only if (G_d, k') is a* yes-*instance of* ParaDdIS(k') *on bipartite graphs.* □

Even if the input graph is restricted to planar bipartite graphs of maximum degree three, DdIS remains intractable for $d \geq 3$. Note that a planar bipartite graph is of course bipartite, and therefore D2IS on planar bipartite graphs is tractable.

Theorem 10 *For every fixed integer $d \geq 3$,* DdIS *is $\mathcal{NP}$-complete even for planar bipartite graphs of maximum degree three.*

proof 11 *We first show the $\mathcal{NP}$-completeness of* D3IS *and then one of the general* DdIS *for $d \geq 4$. Obviously,* DdIS *is in $\mathcal{NP}$ for every $d \geq 3$. To show that* D3IS *is $\mathcal{NP}$-complete, we reduce the $\mathcal{NP}$-complete problem* D2IS *on any cubic planar graph $G_2 = (V_2, E_2)$ to* D3IS *on a new planar bipartite graph $G_3 = (V_3, E_3)$ of maximum degree three.*

Let $V_2 = \{v_1, v_2, \cdots, v_n\}$ and $E_2 = \{e_1, e_2, \cdots, e_m\}$ be vertex and edge sets of the planar graph G_2. We construct the planar bipartite graph G_3 which consists of (i) n vertices, u_1 through u_n, which are associated with n vertices in V_2, v_1 through v_n, respectively, and (ii) m subgraphs, $SG_{3,1}$ through $SG_{3,m}$, which are associated with m edges in E_2, e_1 through e_m, respectively. For every i ($1 \leq i \leq m$), the ith subgraph $SG_{3,i}$ contains three vertices, $w_{i,0}$, $w_{i,1}$, and $w_{i,2}$ and two edges, $\{w_{i,0}, w_{i,1}\}$ and $\{w_{i,1}, w_{i,2}\}$ such that $SG_{3,i}$ forms a path P_2 of length 2. (iii) If $e_i = \{v_j, v_k\} \in E_2$, then we introduce two edges $\{w_{i,0}, u_j\}$ and $\{w_{i,0}, u_k\}$. Note that every simple path $SG_{3,i}$ of length two becomes a single vertex by applying the edge-contraction

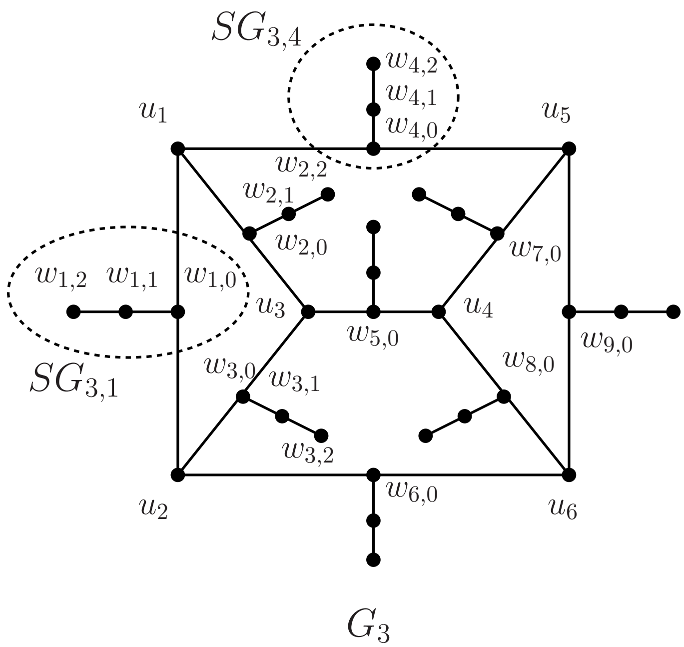


Fig. 4.2 An illustration of the construction when $d = 3$.

twice, and also every path $\langle u_j, w_{i,0}, u_k\rangle$ becomes back an edge $\{u_j, u_k\}$ by applying one edge-contraction for $1 \le i \le m$ and $1 \le j, k \le n$. Namely, the constructed graph G_3 is a minor of the planar graph G_2 and thus it must be planar. The maximum degree is clearly three. The construction can be accomplished in polynomial time. For example, if the cubic planar graph G_2 is the left graph in Figure 4.1, then the reduced graph G_3 is illustrated in Figure 4.2.

For the above construction of G_3, we will show that G_3 has a distance-3 independent set S_3 such that $|S_3| \ge k + m$ if and only if G_2 has a distance-2 independent set S_2 such that $|S_2| \ge k$.

(If part) Suppose that the graph G_2 of D2IS *has a distance-2 independent set $S_2 = \{v_{1^*}, v_{2^*}, \cdots v_{k^*}\}$ in G_2, where $\{1^*, 2^*, \cdots, k^*\} \subseteq \{1, 2, \cdots, n\}$. Then, we select two subsets of vertices $S'_3 = \{u_{1^*}, u_{2^*}, \cdots, u_{k^*}\}$ and $S''_3 = \{w_{1,2}, w_{2,2}, w_{3,2}, \cdots, w_{m,2}\}$ such that $|S'_3| = k$ and $|S''_3| = m$. One can verify that $S_3 = S'_3 \cup S''_3$ is a distance-3 independent set in G_3 since the pairwise distance in S'_3 is at least four, the pairwise distance in S''_3 is at least six, and the distance between $w_{i,2}$ in S''_3 and every vertex in S'_3 is at least three for each i.*

(Only-if part) Conversely, suppose that the graph G_3 has the distance-3 independent set S_3 such that $|S_3| \geq k + m$. First, from each subgraph $SG_{3,i}$ which is the path of length 2, we can select at most one vertex as the distance-3 independent set since its diameter is two. Thus, the maximum size of the distance-3 independent set in $V(SG_{3,1}) \cup V(SG_{3,2}) \cup \cdots \cup V(SG_{3,m})$ is at most m, which means that $|S_3 \cap \{u_1, u_2, \cdots, u_n\}| \geq k$ holds. Let $\{u_{1^*}, u_{2^*}, \cdots, u_{k^*}\}$ be a subset of k vertices in $S_3 \cap \{u_1, u_2, \cdots, u_n\}$. Then, the pairwise distance in the corresponding subset of vertices $\{v_{1^*}, v_{2^*}, \cdots, v_{k^*}\}$ of G_2 is surely at least two, i.e., G_2 has a distance-2 independent set S_2 such that $|S_2| \geq k$. This completes the proof of the $\mathcal{NP}$-hardness of D3IS.

In the following, we give a brief sketch of the ideas to prove the $\mathcal{NP}$-hardness of DdIS *for $d \geq 4$. In the case of* D4IS*, all we have to do is replace the 2-length path $SG_{3,i}$ corresponding to the edge e_i with a 3-length path $SG_{4,i} = (\{w_{i,0}, w_{i,1}, w_{i,2}, w_{i,3}\}, \{(w_{i,0}, w_{i,1}), (w_{i,1}, w_{i,2}), (w_{i,2}, w_{i,3})\})$ for each i. See the left graph in Figure 4.3. In the case of* D5IS*, $SG_{3,i}$ is replaced with $SG_{5,i} = (V(SG_{5,i}), E(SG_{5,i}))$:*

$$V(SG_{5,i}) = \{w^0_{i,0}, w^1_{i,0}, w^2_{i,0}, w_{i,1}, w_{i,2}, w_{i,3}\}$$

$$E(SG_{5,i}) = \{\{\}w^0_{i,0}, w^1_{i,0}\}, \{w^1_{i,0}, w^2_{i,0}\}, \{w^1_{i,0}, w_{i,1}\}, \{w_{i,1}, w_{i,2}\}, \{w_{i,2}, w_{i,3}\}\}.$$

Then, u_j (u_k) corresponding to the vertex v_j (v_k) is connected to $w^0_{i,0}$ ($w^2_{i,0}$) if $e_i = \{v_j, v_k\} \in E_2$ (see the center graph in Figure 4.3). For $d = 6$, we connect one vertex $w_{i,4}$ to the top vertex $w_{i,3}$ of $SG_{5,i}$ (see the right graph in Figure 4.3). Similarly, for a general $d \geq 7$, such a $\perp$-shape subgraph consists of one horizontal path of length $2\lceil \frac{d}{4} \rceil - 2$ and one vertical path of $d - \lceil \frac{d}{4} \rceil$. Since the diameter of $SG_{d,i}$ is less than d, we can select at most one vertex as the distance-d independent set from each subgraph $SG_{d,i}$ as before. Also, if $\{v_i, v_j\} \in E_2$, then $dist_{G_d}(u_i, u_j) < d$; on the other hand if $dist_{G_2}(v_i, v_j) \geq 2$, then $dist_{G_d}(u_i, u_j) = 2 \times 2\lceil \frac{d}{4} \rceil \geq d$. □

4.4 Chordal Graphs

In this section we restrict the instances to chordal graphs. In [18], Gavril shows that D2IS admits an efficient algorithm for chordal graphs:

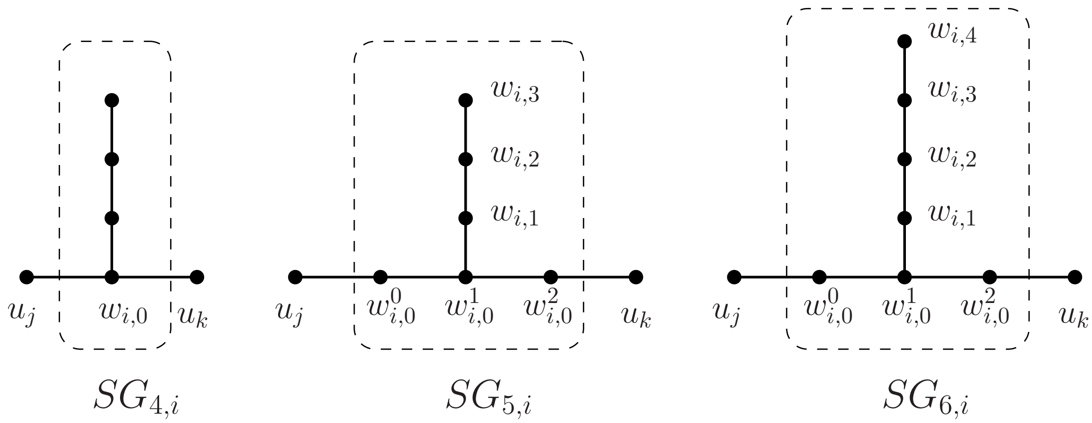


Fig. 4.3 (Left) subgraphs $SG_{4,i}$ for $d = 4$, (Center) $SG_{5,i}$ for $d = 5$, and (Right) $SG_{6,i}$ for $d = 6$.

Lemma 9 ([18]) D2IS *is in $\mathcal{P}$ for chordal graphs.*

Recall that if the dth power graph G^d is interval (trapezoid, or circular-arc, resp.), then the $(d + 1)$th power G^{d+1} is also interval [35] (trapezoid [14], or circular-arc [15], resp.) for any integer $d \geq 1$. The class of chordal graphs *does not* satisfy the closure property under the graph power operation, i.e., the square G^2 of a chordal graph G is not necessarily chordal, but it *does* satisfy the closure property under the graph *odd power* operation:

Lemma 10 ([2, 3]) *Let $d_o \geq 1$ be an odd integer. If G is a chordal graph, then G^{d_o} is also chordal.*

Together with Lemma 9, this yields:

Theorem 11 *For every fixed even integer $d_e \geq 2$,* Dd_eIS *is in $\mathcal{P}$ for chordal graphs.*

proof 12 *Given a chordal graph G, we first construct the odd power graph G^{d_e-1} from G in polynomial time, which must be chordal by Lemma 10. Then, by using a polynomial-time algorithm for* D2IS *in Lemma 9, we can obtain a solution of* Dd_eIS *in polynomial time.* □

For an odd d_o, Dd_oIS is hard:

Theorem 12 *For every fixed odd $d_o \geq 3$, $\mathsf{D}d_o\mathsf{IS}$ is $\mathcal{NP}$-complete for chordal graphs.*

proof 13 *Obviously, $\mathsf{D}d_o\mathsf{IS}$ on chordal graphs is in $\mathcal{NP}$ for every odd $d_o \geq 3$. To show that $\mathsf{D}d_o\mathsf{IS}$ on chordal graphs is $\mathcal{NP}$-complete, we reduce $\mathsf{D2IS}$ on any graph $G_2 = (V_2, E_2)$ to $\mathsf{D}d_o\mathsf{IS}$ on a new chordal graph $G_{d_o} = (V_{d_o}, E_{d_o})$.*

Given the graph $G_2 = (V_2, E_2)$ of $\mathsf{D2IS}$ with n vertices, $V_2 = \{v_1, v_2, \cdots, v_n\}$, and m edges, $E_2 = \{e_1, e_2, \cdots, e_m\}$, we construct the following chordal graph G_{d_o}: (i) We prepare n paths of length $(d_o-3)/2$, $SG_{d_o,1} = \langle u_{1,1}, u_{1,2}, \cdots, u_{1,(d_o-1)/2}\rangle$ through $SG_{d_o,n} = \langle u_{n,1}, u_{n,2}, \cdots, u_{n,(d_o-1)/2}\rangle$, each $SG_{d_o,i}$ of which is corresponding to $v_i \in V_2$, and (ii) m vertices, w_1 through w_m, each w_i of which is corresponding to $e_i \in E_2$. (iii) All the vertices w_1 through w_m are connected such that $G[\{w_1, \cdots, w_m\}]$ forms a clique of m vertices. (iv) If $e_i = \{v_j, v_k\} \in E_2$, then we connect w_i to two vertices $u_{j,1}$ and $u_{k,1}$.

Figure 4.4 illustrates the reduced graph G_7 from G_2 which is illustrated in Figure 4.1 when $d = 7$. The constructed graph G_{d_o} is chordal since all C_4's in the clique graph $G[\{w_1, \cdots, w_m\}]$ have chords and also $G[\{w_1, \cdots, w_m\} \cup \{v_{i,0}\}]$ contains only cycles C_3's for every i. G_{d_o} can be constructed in polynomial time from G_2.

We show that the reduction satisfies that if G_{d_o} has a distance-d_o independent set S_{d_o} such that $|S_{d_o}| \geq k$ if and only in G_2 has a distance-2 independent set S_2 such that $|S_2| \geq k$. In the remaining of this proof, the crucial observations are: (1) The distance between any vertex v in $V_{d_o} \setminus \{u_{1,(d_o-1)/2}, u_{2,(d_o-1)/2}, \cdots, u_{n,(d_o-1)/2}\}$ and another vertex u in $V_{d_o} \setminus \{v\}$ is at most $d_o - 1$. On the other hand, (2) the pairwise distance of any two vertices in $\{u_{1,(d_o-1)/2}, u_{2,(d_o-1)/2}, \cdots, u_{n,(d_o-1)/2}\}$ is at most d_o. The two observations (1) and (2) imply that the distance-d_o independent set S_{d_o} in G_{d_o} must be a subset of outside vertices $\{u_{1,(d_o-1)/2}, u_{2,(d_o-1)/2}, \cdots, u_{n,(d_o-1}$ (3) If v_j and v_k are two endpoints of single edge e_i in G_2, then there must be a path

$$\langle u_{j,(d_o-1)/2}, u_{j,(d_o-3)/2}, \cdots, u_{j,1}, w_i, u_{k,1}, u_{k,2}, \cdots, u_{k,(d_o-1)/2}\rangle$$

by the above reduction rules. Thus, the distance between u_{j,d_o} and u_{k,d_o} in G_{d_o} is $(d_o - 1)/2 \times 2 = d_o - 1$.

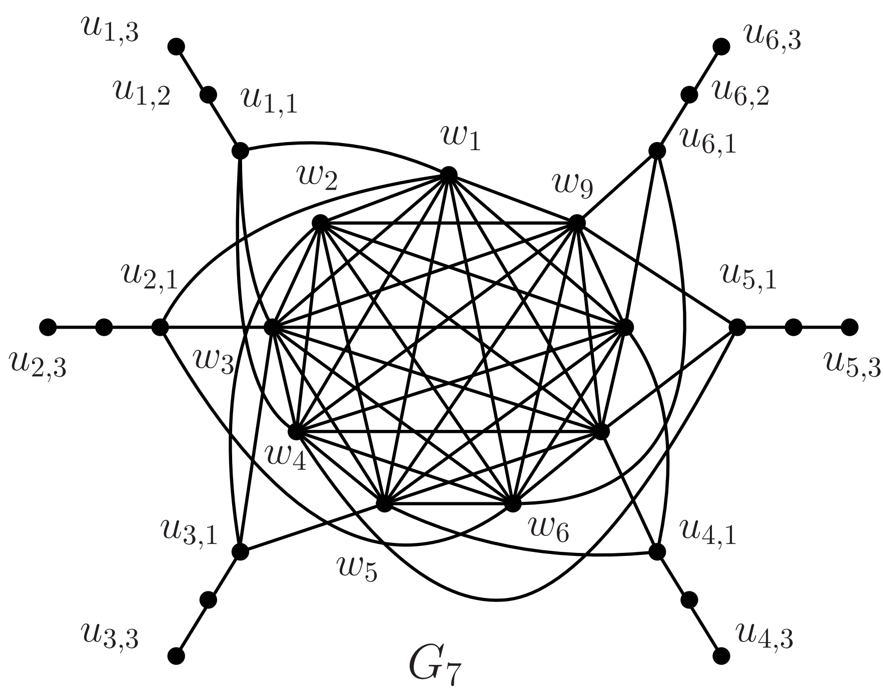


Fig. 4.4 An illustration of the construction when $d = 7$.

(If part) Now suppose that the graph G_2 of D2IS *has a distance-2 independent set $S_2 = \{v_{1^*}, v_{2^*}, \cdots v_{k^*}\}$ in G_2, where $\{1^*, 2^*, \cdots, k^*\} \subseteq \{1, 2, \cdots, n\}$. Then, we select a subset $S_{d_o} = \{u_{1^*,(d_o-1)/2}, u_{2^*,(d_o-1)/2}, \cdots, u_{k^*,(d_o-1)/2}\}$ of size k. It is easy to verify that the pairwise distance in S_{d_o} is exactly d_o.*

(Only-if part) Conversely, suppose that the reduced graph G_{d_o} has the distance-d_o independent set $S_{d_o} = \{u_{1^*,(d_o-1)/2}, u_{2^*,(d_o-1)/2}, \cdots, u_{k^*,(d_o-1)/2}\}$ of size k. Then, the pairwise distance in the corresponding subset of vertices $\{v_{1^*}, v_{2^*}, \cdots, v_{k^*}\}$ of G_2 is surely at least two, i.e., G_2 has a distance-2 independent set S_2 such that $|S_2| \geq k$. □

Corellary 6 D3IS *is $\mathcal{NP}$-complete for split graphs.*

proof 14 *When $d = 3$ in the proof of Theorem 12, the constructed graph G_3 is a split graph since there is a partition of $V(G_3)$ into a clique set $\{w_1, w_2, \cdots, w_m\}$ and an independent set $\{u_{1,1}, u_{2,1}, \cdots, u_{n,1}\}$.* □

Similarly to the previous section, it can be shown that the reduction in the proof of Theorem 12 can preserve the approximation-gap, and also it is an fpt-reduction:

Corellary 7 *For any $\varepsilon > 0$ and fixed odd integer $d_o \geq 3$, it is $\mathcal{NP}$-hard to approximate* $\mathsf{MaxD}d_o\mathsf{IS}$ *to within a factor of $n^{1/2-\varepsilon}$ for chordal graphs.*

proof 15 *The proof is very similar to the proof of Corollary 4. Now, let $OPT'(G_{d_o})$ denote the number of vertices of an optimal solution for a v-vertex input chordal graph G_{d_o} of* $\mathsf{MaxD}d_o\mathsf{IS}$ *for a fixed $d_o \geq 3$. Then, we can show that (1) if $OPT(G_2) \geq g(n)$, then $OPT'(G_{d_o}) \geq g(n)$, and (2) if $OPT(G_2) < \frac{g(n)}{n^{1-\varepsilon}}$ for a positive constant ε, then $OPT'(G_{d_o}) < \frac{g(n)}{n^{1-\varepsilon}}$. Hence the corollary follows from $v = O(n^2)$.* □

Corellary 8 *For every fixed odd integer $d_o \geq 3$,* $\mathsf{ParaD}d_o\mathsf{IS}(k)$ *is $\mathcal{W}[1]$-hard for chordal graphs.*

proof 16 *Let (G_2, k) and (G_{d_o}, k') be the inputs of* $\mathsf{ParaD2IS}(k)$ *and* $\mathsf{ParaD}d_o\mathsf{IS}(k')$ *on chordal graphs, respectively. Then, the reduction in the proof of Theorem 12 satisfies the condition $k' \leq k$.* □

4.5 Concluding Remarks

In the conference version [13] of this chapter we claimed that the reduced graph G_d in the proof of Theorem 9 is *chordal bipartite* and thus $\mathsf{D}d\mathsf{IS}$ on chordal bipartite graphs is $\mathcal{NP}$-hard. However, G_d is not chordal bipartite since it includes an induced cycle of length six or more (for example, actually G_3 in Figure 4.1 contains an induced cycle $\langle u_1, w_1, u_2, w_3, u_3, w_2, u_1 \rangle$ of length six). Therefore, the computational complexity of $\mathsf{D}d\mathsf{IS}$ on chordal bipartite graphs is still open.

Chapter 5

Conclusion

In the chapter 3, we studied the complexity statuses of the r-MaxRIS and r-MaxRICS problems from the viewpoint of graph classes, and analyzed which graph property makes the problems tractable/intractable. We remark that both of our algorithms for graphs with bounded treewidth run in polynomial time even if the degree constraint r is not a fixed constant; see Eqs. (3.2) and (3.3). Furthermore, these algorithms can be easily modified so that they solve more general problems, defined as follows: Given a bounded treewidth graph G and two integers l and u with $l \leq u$, we wish to find a maximum vertex-subset S of G such that every vertex in $G[S]$ is of degree at least l and at most u; as a variant, we may consider the problem which requires $G[S]$ to be connected. Then, these two problems are generalization of r-MaxRIS and r-MaxRICS; consider the case where $l = u = r$.

Furthermore, we studied the Distance-d Independent Set problem, that is a generalization of the Independent Set problem. In the conference version [13] of the chapter 4, we claimed that the reduced graph G_d in the proof of Theorem 9 is *chordal bipartite* and thus DdIS on chordal bipartite graphs is $\mathcal{NP}$-hard. However, G_d is not chordal bipartite since it includes an induced cycle of length six or more (for example, actually G_3 in Figure 4.1 contains an induced cycle $\langle u_1, w_1, u_2, w_3, u_3, w_2, u_1 \rangle$ of length six). Therefore, the computational complexity of DdIS on chordal bipartite graphs is still open.

References

[1] G. Agnarsson, P. Damaschke, M.H. Halldórsson. Powers of geometric intersection graphs and dispersion algorithms. *Discrete Applied Mathematics*, **132**, pp.3–16 (2004)

[2] G. Agnarsson, R. Greenlaw, and M.M. Halldórsson. On powers of chordal graphs and their colorings. *Congr. Numer.*, **144**, pp.41–65 (2000)

[3] R. Balakrishnan and P. Paulraja. Powers of chordal graphs. *Australian Journal of Mathematics*, Series A, **35**, pp.211–217 (1983)

[4] N. Betzler, R. Niedermeier, and J. Uhlmann. Tree decompositions of graphs: saving memory in dynamic programming. Discrete Optimization, vol. 3, pp. 220–229 (2006)

[5] J. R. S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. Graph Theory and Sparse Matrix Computation, vol. 56, pp. 1–29 (1993)

[6] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Computing, vol. 25, pp. 1305–1317 (1996)

[7] A. Brandstädt and V. Giakoumakis. Maximum weight independent sets in hole- and co-chair-free graphs. *Information Processing Letters*, **112**, pp.67–71 (2012)

[8] A. Brandstädt, V.B. Le, J. P. Spinrad. *Graph Classes: A Survey*, SIAM (1999)

[9] K. Cameron. Induced Matchings. *Discrete Applied Math*, 24 (1-3), pp.97–102 (1989)

[10] D. M. Cardoso, M. Kamiński, and V. Lozin. Maximum k-regular induced subgraphs. J. Combinatorial Optimization, vol. 14, pp. 455–463 (2007)

[11] B. Courcelle. Handbook of Theoretical Computer Science (vol. B), Graph rewriting: an algebraic and logic approach, pp. 193–242. MIT Press (1990)

[12] R.G. Downey and M.R. Fellows. Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science* A, **141** (1-2), pp.109–131 (1995)

[13] H. Eto, F. Guo, and E. Miyano. Distance-d Independent Set Problems for Bipartite and Chordal Graphs. In *Proc. COCOA 2012*, pp.234–244 (2012)

[14] C. Flotow. On powers of m-trapezoid graphs. *Discrete Applied Mathematics*, **63**, pp.187–192 (1995)

[15] C. Flotow. On powers of circular arc graphs and proper circular arc graphs. *Discrete Applied Mathematics*, **74**, pp.199–207 (1996)

[16] M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified $\mathcal{NP}$-complete graph problems. *Theoretical Computer Science*, **1**, pp.237–267 (1976)

[17] M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA (1979)

[18] F. Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of chordal graph. *SIAM J. Comput.*, **1**, pp.180–187 (1972)

[19] F. Gavril. Algorithms on circular-arc graphs. *Networks*, **4**, pp.357–369 (1974)

[20] M.C. Golumbic. The complexity of comparability graph recognition and coloring. *Computing*, **18**, pp.199–208 (1977)

[21] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980

[22] F. Harary. *Graph Theory*. Addison-Wesley (1969)

[23] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182 (1), pp.105–142 (1999)

[24] I. A. Stewart. Deciding whether a planar graph has a cubic subgraph is NP-complete. Discrete Mathematics, vol. 126, pp. 349–357 (1994)

[25] I. A. Stewart. Finding regular subgraphs in both arbitrary and planar graphs. Discrete Applied Mathematics, vol. 68, pp. 223–235 (1996)

[26] I. A. Stewart. On locating cubic subgraphs in bounded-degree connected bipartite graphs. Discrete Mathematics, vol. 163, pp. 319–324 (1997)

[27] V. Kann. Strong lower bounds on the approximability of some NPO PB-complete maximization problems. In *Proc. MFCS 1995*, pp.227–236 (1995)

[28] M. Lampis. Algorithmic meta-theorems for restrictions of treewidth. Algorithmica, vol. 64, pp. 19–37 (2012)

[29] V.V. Lozin and M. Milanič. A polynomial algorithm to find an independent set of maximum weight in a fork-free graph. *J. Discrete Algorithms*, **6**, pp.595–604 (2008)

[30] C. Lund and M. Yannakakis. The approximation of maximum subgraph problems. In *Proc. ICALP 1993*, pp.40–51 (1993)

[31] G.J. Minty. On maximal independent sets of vertices in claw-free graphs. *J. Combin. Theory Ser. B*, **28**, pp.284–304 (1980)

[32] O.J. Murphy. Computing independent sets in graphs with large girth. *Discrete Applied Mathematics*, **35**, pp.167–170 (1992)

[33] Y. Orlovich, G. Finke, V. Gordon, and I. Zverovich. Approximability results for the maximum and minimum maximal induced matching problems. Discrete Optimization, vol. 5, pp. 584–593 (2008).

[34] S. Poljak. A note on stable sets and coloring of graphs. *Comment. Math. Univ.* Carolin. 15 pp.307–309 (1974)

[35] A. Raychaudhuri. On powers of interval and unit interval graphs. *Congr. Numer.*, **59**, pp.235–242 (1987)

[36] V.V. Vazirani. *Approximation Algorithms*, Springer (2003)

[37] D. Zuckerman. Linear degree extractors and the inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, **3**(1), pp.103–128 (2007)